# Instruction manual Vol. 2

**Honeywell**

COMPUTER CONTROL DIVISION

## Series 16 general purpose I/C digital computer

WEISE

Doc. No. 130071621

M-967

# Instruction Manual

# DDP-516

# GENERAL PURPOSE COMPUTER

## Volume II

September 1966

**Honeywell**

COMPUTER CONTROL DIVISION

COPYRIGHT 1966 by Honeywell Inc., Computer Control Division, Framingham, Massachusetts. Contents of this publication may not be reproduced in any form in whole or in part, without permission of the copyright owner. All rights reserved.

Printed in U.K. COPIA PRODUCTUM, HARROW.

CONTENTS

VOLUME II


| | | Page |
|---|---|---|
| Chapter I | Introduction | 1-1 |
| | Signal Mnemonics | 1-2 |
| | Table 1-1 Symbols and Definitions | 1-2 |
| | Table 1-2 Function Index | 1-3 |
| Chapter II | Flow Charts/Instruction Analyses | 2-1 |
| | F-Cycle | 2-4 |
| | I-Cycle | 2-6 |
| Memory Reference | | |
| | ADD - Add | 2-8 |
| | ANA - Logical AND | 2-10 |
| | CAS - Compare | 2-12 |
| | ERA - Exclusive OR | 2-14 |
| | IMA - Interchange Memory and A | 2-16 |
| | IRS - Increment, Replace, and Skip | 2-18 |
| | JMP - Unconditional Jump | 2-20 |
| | JST - Jump and Store Location | 2-22 |
| | LDA - Load A | 2-24 |
| | LDX - Load Index Register | 2-26 |
| | STA - Store A | 2-28 |
| | STX - Store Index Register | 2-30 |
| | SUB - Subtract | 2-32 |
| Generic | | |
| | ACA - Add C to A | 2-34 |
| | ADA - Add ONE to A | 2-36 |
| | CAL - Clear Left Half | 2-38 |
| | CAR - Clear Right Half | 2-40 |
| | CAS - Complement A Sign | 2-42 |
| | CMA - Complement A | 2-44 |
| | CRA - Clear A-Register | 2-46 |
| | CSA - Copy Sign and Set Sign Plus | 2-48 |
| | ENB - Enable Program Interrupt | 2-50 |
| | HLT - Halt | 2-52 |
| | IAB - Interchange A and B | 2-54 |
| | ICA - Interchange Halves | 2-56 |
| | ICL - Interchange and Clear Left Half | 2-58 |

CONTENTS (Cont)


| | Page |
|---|---|
| ICR - Interchange and Clear Right Half | 2-60 |
| INH - Inhibit Program Interrupt | 2-62 |
| INK - Input Keys | 2-64 |
| NOP - No Operation | 2-66 |
| OTK - Output Keys | 2-68 |
| RCB - Rese C to ZERO | 2-70 |
| SCB - Set C to ONE | 2-72 |
| COMMON SKIP | 2-74 |
| SSM - Set Sign Minus | 2-76 |
| SSP - Set Sign Plus | 2-78 |
| TCA - Two's Complement A | 2-80 |
| Input-Output | |
| INA - Input to A | 2-82 |
| OCP - Output Control Pulse | 2-84 |
| OTA/SMK - Output from A/Set Mask | 2-86 |
| SKS - Skip if Sense Line Set | 2-88 |
| Shift | |
| COMMON SHIFT ENTRY | 2-90 |
| ALR - Left Logical Rotate | 2-90 |
| ALS - Arithmetic Left Shift | 2-92 |
| ARR - Logical Right Rotate | 2-94 |
| ARS - Arithmetic Right Shift | 2-96 |
| LGL - Logical Left Shift | 2-98 |
| LGR - Logical Right Shift | 2-100 |
| LLL - Long Left Logical Shift | 2-102 |
| LLR - Long Left Rotate | 2-104 |
| LLS - Long Arithmetic Left Shift | 2-106 |
| LRL - Long Right Logical Shift | 2-108 |
| LRR - Long Right Rotate | 2-110 |
| LRS - Long Arithmetic Right Shift | 2-112 |
| COMMON SHIFT EXIT | 2-113 |
| Table 2-1 Generating 0PGJS+ for Specific Skip Instructions | 2-75 |

DDP-516 GENERAL PURPOSE COMPUTER

3615 DDP-516 General Purpose Computer

# SECTION I
# INTRODUCTION

This volume contains flow charts and analyses of all DDP-516 instructions. Separate coverage is provided for the memory reference fetch cycle. The instructions are presented in groups (i.e., load and store, arithmetic, logical, etc.). For convenient reference, the mnemonic, number of cycles, and Op Code appear on each page. Symbols and abbreviations used in the flow charts are defined in Table 1-1. A function index is provided to be used as an aid in determining function sources and definitions, (Table 1-2).

Signal Mnemonics

The electrical characteristics of the computer circuits are called: passive (+6 vdc) and active (0 vdc). The logical functions are true (logic ONE) and false (logic ZERO). In general there are two ways of relating the electrical characteristic and logical function for each signal mnemonic:

a. An assertion signal (for example, A01FF+) is logically true when it is at +6 vdc; and is logically false when it is at 0 vdc.

b. A negation signal (for example, A01FF-) is logically true when it is at 0 vdc; and is logically false when it is at +6 vdc.

A particular signal mnemonic can be labeled assertion or negation arbitrarily. The signal mnemonic convention uses the sixth character to specify assertion or negation.

Table 1-1.
Symbols and Definitions

| Symbol | Definition |
|---|---|
| $\wedge$ or $\cdot$ | Logical AND |
| $\vee$ | Logical OR |
| $\veebar$ or $\oplus$ | Exclusive OR |
| $\rightarrow$ | Replaces |
| $\rightleftarrows$ | Is exchanged with |
| $\neg\downarrow$ | Is discarded |
| ( ) | Contents of a hardware register |
| [EA] | Contents of a core location specified by the effective operand address |
| A | A-register |
| ADB | Address bus |
| B | B-register |
| C | Carry flip-flop (CB1TF) |
| E | E-register |
| EA | Effective operand address; the address from which the operand will be obtained |
| F | Flag (indirect address indicator) or F-register |
| INB | Input bus |
| IW | Instruction word |
| M | M-register |
| N | Two's complement of number of shifts to be performed |
| OTB | Output bus |
| P | P-register |
| X | X-register |

Table 1-2.
Function Index

| Mnemonic | LBD No. (Signal Source) | Definition |
|---|---|---|
| 0 x x PA | 101-116-C5 | Adder input OR gates (G & H), bits 1 through 16 |
| 0CPLS | 134-K10 | Output control pulse |
| 0CPRC | 147-F3 | Output control pulse, real-time clock |
| 0PG00 | 120-F1 | Op group 00 |
| 0PG3C | 120-F9 | Op group for three cycle instructions (JST v IRS v CAS v IMA v LDX v DOUBLE PRECISION) |
| 0PGAA | 127-E9 | Op group, A-cycle utilization of A-register. |
| 0PGDP | 124-F11 | Op group, double precision arithmetic operation (ADD v SUB v LDA v STA) |
| 0PGJS | 129-J11 | Op group, jump or skip (JMP v JST v IRS v (SKIP ENABLED)) |
| 0PGMD | 123-B6 | Op group, multiply or divide (MPY v DIV) |
| 0PGNS | 125-B11 | Op group, negative sum. Control ENSHL/ ENSLL for subtractive process, (SUB v IRS v CASAM) |
| 0PGSM | 128-C10 | Op group, sum to M-register control (STA v IMA v LDX v STX) |
| 0PGWR | 126-E6 | Op group, write/read control (STA v IMA v LDX v STX v IRS v JST) |
| 0TB x x | 138-XX | Output bus, bits 1 through 16 |
| 0TPMA | 240-H6 | Output transfer pulse, DMA option |
| 10GRP | 120-K5 | Op group, input/output group |
| 10PLS | 134-J9 | Input/output pulse |
| 1AD61 | 135-C3 | Interrupt address 61 (RTC location 00061) |
| 1AD63 | 135-C5 | Interrupt address 63 (Standard interrupt location 00063) |
| 1ADX x | 135-C4 | Interrupt address, location 00XX1 through 00XX7 |
| 1CYEF | 119-G8 | I-cycle early flip-flop. (Indirect address) |
| 1CYLF | 119-K7 | I-cycle late flip-flop. (Indirect address) |
| 1CYS1 | 263-E7 | I-cycle and sector zero not selected |
| 1G x x x | 200-XX | Interrupt address greater than 063 or 067 or 077 or 107 or 117 or 127 or 137 (as applicable) |
| 1MA0P | 120-F5 | IMA op code |
| 1NB x x | 101-116-D5 | Input bus, bits 1 through 16 |
| 1NCSC | 126-L3 | Increment shift counter if not $00_8$ |
| 1ND x x | 132-XX | Console indicator lamp drivers, bits 1 through 16 |

Table 1-2. (Cont)
Function Index

| Mnemonic | LBD No. (Signal Source) | Definition |
|---|---|---|
| INHPF | 135-F11 | Inhibit power failure interrupt |
| INK0P | 122-L8 | INK op code |
| INSTR | 241-L2 | Input strobe, DMA option. |
| IRS0P | 120-K3 | IRS op code |
| IYB x x | 223-XX, 135-XX | Input-to-Y bus, bits 2 through 16. DMC option, and program interrupt |
| A x x FF | 101-116-J5 | A-register flip-flops, bits 1 through 16. |
| A00FF | 124-L4 | A00 flip-flop |
| A0QM1 | 124-J8 | A00FF equals M01FF |
| A1QA2 | 126-B3 | A01FF equals A02FF, normalize signal |
| ACKA x | 241-B4 | Acknowledge for DMA channel 1 through 4 |
| ACKPF | 135-B3 | Acknowledge for power failure interrupt |
| ACKRC | 135-B4 | Acknowledge for real-time clock |
| ACT x x | 244-XX | DMA address count bits 1 through 16 |
| ACYEF | 119-F4 | A-cycle early flip-flop |
| ACYLF | 119-L4 | A-cycle late flip-flop |
| ACYNX | 129-B1 | A-cycle is next main frame cycle |
| ADB x x | 138-XX | Address bus, bits 7 through 16 |
| ADD0P | 120-F2 | ADD op code |
| ANA0P | 120-F1 | ANA (logical AND to A)op code |
| AZER0 | 126-L9 | A-register equals zero, bits 1 through 16. |
| AZZZZ | 125-L10 | A general control flip-flop |
| B x x FF | 101-116-J1 | B-register flip-flop, bits 1 through 16. |
| BANK x | 145-C2 | Memory bank select A, B, C, or D |
| BR1CY | 134-C7 | BREAK and I-cycle |
| BREAK | 134-K1 | BREAK flip-flop. Set by program interrupt, RTC, memory increment, or DMC operation, etc. |
| BRREQ | 134-E1 | BREAK request (sync-in to BREAK operation) |
| CAS0P | 120-H4 | CAS op code (compare A and SKIP) |
| CASAM | 126-L4 | A01FF equals M01FF in CAS instruction |
| CB1TF | 124-L2 | C-bit flip-flop (arithmetic overflow, etc.) |
| CHEN x | 241-B9 | Channel enable for DMA option (channels 1 through 4) |
| CHSL x | 241-D3 | Channel select for DMA option (channels 1 through 4) |
| CLA1L | 130-H11 | Clear A-register bit 1 (clock level) |
| CLAMP | 141-J7 | Clear A-register with manual pushbutton switch |
| CLATL | 122-F7 | Clear A-register, total (clock-level) |
| CLATR | 122-H7 | Clear A-register, total (clock- reset) |

Table 1-2. (Cont)
Function Index

| Mnemonics | LBD No. (Signal Source) | Definition |
|---|---|---|
| CLBMP | 141-F7 | Clear B-register with manual pushbutton switch |
| CLBTR | 123-J7 | Clear B-register, total (clock-reset) |
| CLCHS | 241-J5 | Clear DMA channel select flip-flops (clock-reset) |
| CLDTR | 125-H5 | Clear D-register to ONEs, total (clock-reset) |
| CLETR | 125-H2 | Clear E-register to ONEs, total (clock-reset) |
| CLFTL | 125-H8 | Clear F-register, total (clock-level) |
| CLLTR | 241-L10 | Clear L-register, DMA option, total (clock-reset) |
| CLMMP | 141-B7 | Clear M-register with manual pushbutton switch |
| CLMTR | 128-K8 | Clear M-register, total (clock-reset) |
| CLP1L | 134-L2 | Clear program interrupt and memory increment request lines |
| CLPMP | 141-C7 | Clear P- and Y-registers with manual push-button switch |
| CLPTR | 129-J10 | Clear P-register, total (clock-reset) |
| CLRDR | 126-L1 | Clear D-register when master clock oscillator has stopped |
| CLRF5 | 134-E7 | Clear bit 5 of F-register on program interrupt |
| CLSEX | 134-K5 | Clear single execute, (program interrupt and memory increment) request flip-flops |
| CLSZR | 262-H9 | Clear sector zero relocation |
| CLXTR | 128-F8 | Clear X-register, total (clock-reset) |
| CLYTR | 129-J3 | Clear Y-register, total (clock-reset) |
| CLZTL | 241-L5 | Clear Z-register, (DMA option) total (clock-level) |
| CMK09 | 122-L7 | Clear mask flip-flops for OTK instruction |
| CMKXX | 134-K7 | Clear mask, I/O control pulse. |
| D00DJ | 130-D1 | Fictitious D00 bit to take care of right shift end effects |
| D00FF | 130-H9 | D00 control flip-flop, extension of D-register. |
| D$\underline{x}\,\underline{x}$FF | 101-116-F8 | D-register flip-flops, bits 1 through 16 |
| D0GFF | 124-L7 | Divide termination control flip-flop for arithmetic option |
| D17DJ | 130-F3 | Fictitious D17 bit (end effect) of left shift |
| D1LA $\underline{x}$ | 245-F8 | Device interrupt line, DMA channels 1 through 4 |
| D1QAZ | 123-B3 | D01FF equals AZZZZ flip-flop |
| D1V0P | 120-F5 | DIV op code |
| DALEN | 224- | DMC address line enable |
| DCY2X | 221-D10 | DMC cycle 2 |

Table 1-2. (Cont)
Function Index

| Mnemonics | LBD No. (Signal Source) | Definition |
|---|---|---|
| DCY3X | 221-K5 | DMC cycle 3 |
| DCYXX | 221-B10 | DMC cycle 1, 2, and 3 |
| DG0NE | 126-D9 | D-register equals zero, bits 1 through 16. |
| DMA1N | 259-XX | DMA input mode |
| DMACY | 241-L7 | DMA cycle |
| DMARQ | 241-B5 | DMA request |
| DMAWR | 241-G9 | DMA write/read control |
| DMCCY | 138-B11 | DMC cycle 1, 2, 3, and 4 |
| DMCRQ | 220-F6 | DMC request |
| DMCRR | 221-F2 | DMC reset ready line |
| DMCWR | 221-L2 | DMC write/read control |
| DP0LX | 124-J11 | Double precision or LDX op code |
| DPM0D | 124-B10 | Double precision mode flip-flop |
| DRL1N | 143-B4 | Device ready line |
| DSPL0 | 141-D7 | Display operating indicators |
| DSPLA | 141-H7 | Display A-register |
| DSPLB | 141-E7 | Display B-register |
| DSPLM | 141-A7 | Display M-register |
| DSPLX | 141-K7 | Display X-register |
| DSPLY | 141-B7 | Display Y-register |
| E x x FF | 101-116-L3 | E-register flip-flops, bits 1 through 16. |
| E01NS | 119-E1 | End of instruction |
| E0ADJ | 130-D9 | Fictitious E-register bit for shift end effect |
| E0BDJ | 130-F1 | Fictitious E-register bit for shift end effect |
| E0CDJ | 130-D8 | Fictitious E-register bit for shift end effect |
| E0DDJ | 130-D11 | Fictitious E-register bit for shift end effect |
| E0Y16 | 124-J9 | Enable zero to Y-register, bit 16. |
| E1CHL | 125-F10 | Enable ONE to shift counter bit 11 (clock-level) |
| E1CTS | 125-J8 | Enable ONEs to shift counter, total (clock-set) |
| E1DTS | 125-L6 | Enable input bus to D-register, bits 1 through 16 (clock-set) |
| E1K17 | 127-L4 | Enable 1 carry-in from fictitious bit 17 (end inject carry) |
| E1YHS | 129-L7 | Enable IYB, bits 2 through 9, to Y-register (clock-set) |
| E1YLS | 129-L6 | Enable IYB, bits 10 through 16, to Y-register (clock-set) |
| E40SC | 121-C2 | Enable 40 (octal) to shift counter. Setup for NRM instruction |
| E57SC | 124-J3 | Enable 57 (octal) to shift counter. Setup for DIV instruction |

Table 1-2. (Cont)
Function Index

| Mnemonics | LBD No. (Signal Source) | Definition |
|---|---|---|
| E70SC | 124-G3 | Enable 70 (octal) to shift counter. Setup for MPY instruction |
| EASTL | 127-L1 | Enable A-register to sum network, bits 1 through 16 (clock-level) |
| EBETS | 125-L1 | Enable B-register to E-register, bits 1 through 16 (clock-set) |
| ECETS | 125-L3 | Enable shift counter to E-register, bits 11 through 16 (clock-set) |
| EDA1L | 130-H10 | Enable D-register to A-register, bit 1 (clock-level) |
| EDAHS | 122-L1 | Enable D-register to A-register, bits 1 through 8 (clock-set) |
| EDALS | 122-L2 | Enable D-register to A-register, bits 9 through 16 (clock-set) |
| EDBTS | 123-L2 | Enable D-register to B-register, bits 1 through 16 (clock-set) |
| EDM $\underline{x}\,\underline{x}$ | 101-116-G8 | Enable D-register to M-register, bits 1 through 16 (DJs with ESM__ gates) |
| EDMAY | 241-L3 | Enable DMA bus (address count or Z-register) to Y-register, bits 1 through 16 (clock-set) |
| EDMTS | 128-H11 | Enable D-register to M-register, bits 1 through 16 (clock-set) |
| EDPTL | 129-G9 | Enable D-register to P-register (clock-level) |
| EDPTS | 129-L9 | Enable D-register to P-register, bits 1 through 16 (clock-set) |
| EDYTS | 129-L2 | Enable D-register to Y-register, bits 1 through 16 (clock-set) |
| EEALS | 122-K4 | Enable E-register to A-register, bits 11 through 16 (clock-set) |
| EEATS | 122-L4 | Enable E-register to A-register, bits 1 through 10 (clock-set) |
| EMC $\underline{x}\,\underline{x}$ | 121-X11 | Enable M-register to shift counter, bits 11 through 16, respectively. |
| EMCTL | 125-J11 | Enable M-register to shift counter, bits 11 through 16 (clock-level) |
| EMFTL | 128-K4 | Enable M-register to F-register, bits 3 through 6 (clock-level) |
| EMSHL | 127-L8 | Enable M-register to sum network, bits 1 through 7 (clock-level) |
| EMSLL | 127-L10 | Enable M-register to sum network, bits 8 through 16 (clock-level) |
| EMXTS | 128-H7 | Enable M-register to X-register, bits 1 through 16 (clock-set) |
| ENSHL | 127-L7 | Enable M-register (negation) to sum network, bits 1 through 7 (clock-level) |
| ENSK1 | 150-K2, 3 | Selection Sink Enable Pulse |

Table 1-2. (Cont)
Function Index

| Mnemonics | LBD No. (Signal Source) | Definition |
|---|---|---|
| ENSLL | 127-L5 | Enable M-register (negation) to sum network, bits 8 through 16 (clock-level) |
| ENTRA | 141-H4 | Enter A-register (manual console op) |
| ENTRB | 141-E4 | Enter B-register (manual console op) |
| ENTRM | 141-A4 | Enter M-register (manual console op) |
| ENTRP | 141-B4 | Enter P-register (manual console op) |
| ENYSW | 150-K4, 5 | Digit Selection Switch Enable Pulse |
| EPSLL | 128-H4 | Enable P-register to sum network bits 3 through 16 and enable M-register bits 1 and 2 to sum network (clock-level) |
| EPYTS | 129-L4 | Enable P-register to Y-register, bits 1 through 16 (clock-set) |
| ERA0P | 120-K1 | ERA op code (exclusive OR to A) |
| ERLAX | 259-XX | End of range for DMA |
| ERLXX | 222-L11 | End of range line for DMC |
| ESDTS | 125-L4 | Enable adder sum to D-register, bits 1 through 16 (clock-set) |
| ESMTS | 128-H10 | Enable sum network input to M-register, bits 1 through 16 (clock-set) |
| ETAHS | 122-L5 | Enable transposed D-register, bits 9 through 16 into A-register, bits 1 through 8 (clock-set) |
| ETALS | 122-L6 | Enable transposed D-register bits 1 through 8 into A-register, bits 9 through 16 (clock-set) |
| EXSTL | 128-K5 | Enable X-register to sum network, bits 1 through 16 (clock-level) |
| EXTMD | 136-H1 | Extended addressing mode flip-flop |
| EYSHL | 128-K3 | Enable Y-register to sum network, bits 1 through 7 (clock-level) |
| EYSLL | 128-K2 | Enable Y-register to sum network, bits 8 through 16 (clock-level) |
| EYZTL | 241-L6 | Enable Y-register to Z-register, bits 1 through 16 (clock-level) |
| EZYTL | 241-J7 | Enable Z-register to Y-register via ACT__+ lines, bits 1 through 16 (clock-level) |
| F01CY | 119-D7 | F or I cycle control flip-flops are set |
| F0 x FF | 120-DX | Function register, bits 3 through 6 |
| FCX00 | 134-J11 | Function control indicating address bus lines 08, 09, 10, and 14 are all zero |
| FCYEF | 119-F10 | F-cycle early flip-flop |
| FCYLF | 119-K10 | F-cycle late flip-flop |
| FCYM2 | 136-F4 | F-cycle/M02FF (tag store flip-flop for extended addressing mode) |
| FCYS0 | 127-F5 | F-cycle sector zero specified in memory reference instruction addressing |

Table 1-2. (Cont)
Function Index

| Mnemonics | LBD No. (Signal Source) | Definition |
|---|---|---|
| G x x DJ | 101-116-B4 | Adder network input (A v X), bits 1 through 16, respectively |
| GEN0A | 120-K11 | Generic operation, class A |
| GEN0B | 120-K10 | Generic operation, class B |
| GEN0P | 120-C11 | Generic operation |
| H x x DJ | 101-116-B9 | Adder network input (M v P v Y v M -), bits 1 through 16, respectively |
| HOLDM | 128-J8 | Hold M-register contents (inhibit CLMTR-) |
| JAMKN | 127-J4 | Jam carry network (suppress carries) |
| JMP0P | 120-J1 | JMP op code |
| JST0P | 120-K4 | JST op code |
| L x x FF | 242-243-XX | L-register flip-flops, bits 1 through 16 (DMA option) |
| L0B x x | 242-243-XX | L-register output bus, bits 1 through 16 (DMA option) |
| L17XX | 133-XX | Fictitious L17 for parity injection |
| L1B x x | 245-XX | L-register input bus, bits 1 through 16 |
| LDA0P | 120-F1 | LDA op code |
| LMPRN | 132-C3 | Lamp, run indicator |
| LMR x x | 153-161-E5, 11 | Selection Switch Discharge Resistor Line |
| LSX0P | 120-H5 | Load/store X-register op code |
| LXACY | 263-D9 | Load X-register/A-cycle |
| M x x FF | 101-116-J8 | M-register, bits 1 through 16 |
| M01ML | 262-K7 | M01 control from memory lockout |
| M02DJ | 136-H5 | M02 DJ gates for extended addressing |
| M17FF | 133-K10 | Parity bit flip-flop of memory parity option |
| M5G4G | 123-F1 | Minterm control in DP and IAB instructions |
| MACYL | 128-E1 | Multiply control in A-cycle |
| MADFF | 124-L5 | Multiply and divide control flip-flop |
| MAST0 | 141-E9 | Memory access, store mode |
| MBSYL | 150-B2 | End of Memory Busy pulse |
| MCRST | 118-G1 | Master clock, reset phase |
| MCSET | 118-F3 | Master clock, set phase |
| MCTLG | 118-G2 | Master clock, timing level generator phase |
| MDA2A | 123-D3 | MPY/DIV option, A-cycle, TL2, control minterm A |
| MDA2C | 123-G10 | MPY/DIV option, A-cycle, TL2, control minterm C |
| MDG2E | 123-D1 | MPY/DIV option, timing level 2, control minterm E |

Table 1-2. (Cont)
Function Index

| Mnemonics | LBD No. (Signal Source) | Definition |
|---|---|---|
| MDG4D | 123-G9 | MPY/DIV option, timing level 4, control minterm D |
| MD01X to MD17X | 153 to 161 C4, 5 | Memory data output signals |
| MDSLA | 122-B9 | MPY/DIV option, shift left A-register |
| MDSRA | 123-E9 | MPY/DIV option, shift right A-register |
| MEL0V | 262-K3 | Memory lockout violation flip-flop |
| MEMAC | 141-G8 | Memory access mode control |
| MEMC1 | 126-J11 | Memory cycle initiate |
| MFG2E | 122-F11 | Main frame, general cycle, TL2, minterm E |
| MM x x x | 142-XX | Memory to M-register data bits 1 through 16, from memory bank A through D and combinations |
| MPEFF | 133-J7 | Memory parity error flip-flop |
| MPY0P | 120-F5 | MPY op code |
| MSTCL | 141-B10 | Master clear (over all initialization) |
| N x x PA | 101-116-C9 | Adder network input AND gates (G ∧ H), bits 1 through 16, respectively |
| N0P10 | 125-D10 | I/O no operation. Device not ready |
| N0ST0 | 262-D3 | No store. Protected memory area is being accessed. |
| NRM0P | 128-D1 | NRM op code |
| P x x FF | 101-116-J11 | P-register flip-flops, bits 1 through 16 |
| P02BS | 136-D5 | P02 storage flip-flop for bank control extended addressing. |
| P1L00 | 133-L6 | Program interrupt line 00 (standard) |
| P1REQ | 143-XX | Program interrupt request |
| P1SEX | 134-C2 | Program interrupt or single execute request |
| PARCK | 143-B4 | Parity check indicator from I/O |
| PERM1 | 134-E4 | Permit interrupt flip-flop |
| PF1NT | 135-B8 | Power failure interrupt signal from sensing PAC |
| PFHLT | 141-K8 | Power failure halt mode control |
| PM1ND | 136-K1 | Previous mode indicator flip-flop for extended-addressing mode |
| PR0T x | 264-XX | Accessing protected sector 0 through 3<br>00-07 or 40-47<br>10-17 or 50-57<br>20-27 or 60-67<br>30-37 or 70-77 |
| R x x PA | 101-116-C9 | Adder network or gates (G v H-), bits 1 through 16, respectively |

Table 1-2. (Cont)
Function Index

| Mnemonics | LBD No. (Signal Source) | Definition |
|---|---|---|
| RCYF1 | 150-E1 | Read cycle flip-flop |
| RC7SW | 150-H5 | Digit selection switch read timing pulse |
| RDATA | 150-E5 | Memory data input disable pulse |
| READY | 141-J10 | Prestart signal from console pushbutton switch |
| REM0K | 123-B1 | Remainder OK to terminate divide |
| RENSK | 150-E4 | Selection Sink enable read timing pulse |
| RESTR | 262-F5 | Restricted mode flip-flop for memory lockout option |
| RPTT2 | 126-G5 | Repeat TL2 timing level |
| RRCXX | 126-L6 | Read-regenerate cycle control to memory |
| RRL1N | 134-K8 | Reset ready line signal to I/O device |
| RSKA1 | 150-E2 | Selection sink read command timing pulse |
| RTCAD | 147-F9 | Real-time clock address decoder |
| RTCLK | 147-H2 | Real-time clock service request |
| RUNFF | 126-J2 | RUN control flip-flop |
| RUNMD | 141-D10 | RUN mode console switch control |
| RXSWA | 150-E2 | X-Selection switch read command pulse |
| RYSWA | 150-E3 | Digit selection switch read command timing pulse |
| S x x C x | 117-XX | Sum network outputs from carry gating, bits 1 through 16, respectively |
| SAM1N | 240-H3 | Sense amplifier to M-register inhibit. DMA memory control |
| SC x x F | 121-XX | Shift counter flip-flops. Bits 11 through 16. |
| SCQ70 | 121-L1 | Shift counter equals octal 70 |
| SCQ77 | 121-L1 | Shift counter equals octal 77 |
| SCZR0 | 121-L3 | Shift counter equals zero |
| SCZR1 | 121-L2 | Shift counter equals zero or minus one (77) |
| SDARS | 123-L11 | Shift (double) A-register, right shift (clock-set) |
| SDBRS | 123-L10 | Shift (double) B-register, right shift (clock-set) |
| SDB x x | 130-C6 | End effects for SDBRS to B-register, bit 1 and bit 2 |
| SDR x x | 140-X7 | Set data register, bits 1 through 16, from console pushbutton switches |
| SEC x x | 272-xx | Sector (00-77) is unprotected in memory lockout option |
| SENS x | 140-X10 | Sense switches on console, 1 through 4, respectively |
| SETA0 | 125-J9 | Set A00FF control signal |

Table 1-2. (Cont)
Function Index

| Mnemonics | LBD No. (Signal Source) | Definition |
|---|---|---|
| SETAZ | 125-J9 | Set AZZZZ control signal |
| SETF5 | 134-H5 | Set bit 5 of F-register for memory increment |
| SETTA | 240-G11 | Set TLA timing level flip-flop in DMA option |
| SEX00 | 135-B6 | Standard interrupt priority network flip-flop |
| SEXLV | 135-B5 | Memory lockout violation interrupt priority network flip-flop |
| SEXPF | 135-B3 | Power failure interrupt priority network flip-flop |
| SEXRC | 135-B3 | Real-time clock service request priority network flip-flop |
| SEXRQ | 200-XX | Single execute memory increment request |
| SHA0P | 120-K8 | Shift A-register op code |
| SHASC | 120-K7 | Shift A-register and shift counter does not equal zero |
| SKGRP | 120-K6 | Skip group op code |
| SLATS | 122-L10 | Shift left to A-register (clock-set) |
| SLBTS | 123-L4 | Shift left to B-register (clock-set) |
| SMK01 | 134-L11 | Set mask group No. 1 (standard devices) |
| SMK09 | 122-J9 | Set mask group No. 9 (OTK instruction) |
| SMKXX | 134-E10 | Set mask general output strobe |
| SMP1L | 135-A10 | Sample program interrupt lines |
| SPM0D | 135-H9 | Single pulse mode control (service jumper) |
| SRATS | 122-L11 | Shift right to A-register (clock-set) |
| SRBTS | 123-L9 | Shift right to B-register (clock-set) |
| SRSTL | 128-H1 | Shift right from A-register to sum network (clock-level) |
| STA0P | 120-F2 | STA op code |
| START | 141-J9 | Start pushbutton switch release |
| STCHN | 241-L1 | Set channel enable flip-flops in DMA option |
| STCHS | 241-J6 | Set channel select flip-flops in DMA option |
| STEPP | 141-G10 | Step P-register in MEMAC mode |
| STEXT | 136-H9 | Set extended addressing mode |
| STRB1 | 150-G6, 7 | Sense amplifier strobe pulse |
| SUB0P | 120-F1 | SUB op code |
| SW01 to SW17 | (Core Stack) | Sense amplifier core stack sense winding inputs |
| TACFF | 240-C7 | Timing level mid-TLA through mid-TLC (DMA option) |
| TBDFF | 240-C10 | Timing level mid-TLB through mid-TLD (DMA option) |
| TL13F | 118-D5 | Timing level mid-TL1 through mid-TL3 |

Table 1-2. (Cont)
Function Index

| Mnemonics | LBD No. (Signal Source) | Definition |
|---|---|---|
| TL1FF | 118-J11 | Timing level TL1 |
| TL23F | 118-D8 | Timing level mid-TL2 through mid-TL3 |
| TL24F | 118-D11 | Timing level mid-TL2 through mid-TL4 |
| TL2FF | 118-J8 | Timing level TL2 |
| TL3FF | 118-J6 | Timing level TL3 |
| TL4FF | 118-J3 | Timing level TL4 |
| TLATE | 118-B8 | Timing level, late, TL2 and TL3 inclusive |
| TLAFF | 240-F11 | Timing level A in DMA option |
| TLBFF | 240-F9 | Timing level B in DMA option |
| TLCFF | 240-F6 | Timing level C in DMA option |
| TLDFF | 240-F4 | Timing level D in DMA option |
| TLEFF | 240-F2 | Timing level E in DMA option |
| VAC00 | 148-E9 | 0 vac, transformer center tap |
| VAC03 | 148-E9 | 3 vac, PFI transformer |
| VDC06 | | -6 vdc |
| WCYF1 | 150-E7 | Write cycle flip-flop output |
| WENSK | 150-E9 | Selection sink enable write timing pulse |
| WR1NH | 126-D8 | Write inhibit |
| WSKA1 | 150-E10 | Write sink activate |
| WT7SW | 150-H4 | Digit selection switch write timing pulse |
| WXSWA | 150-E11 | X-selection switch write command pulse |
| WYSWA | 150-E9 | Digit selection switch write command timing pulse |
| X x x FF | 101-116-L8 | X-register flip-flops, bits 1 through 16 |
| XD01 to XD32 | 152-XX | X-selection switch outputs |
| XB01 to XB16 | 152-XX | X-selection sink outputs |
| Y x x FF | 101-116-L10 | Y-register flip-flops, bits 1 through 16 |
| Y04XX to Y16XX | 101, 116-L10 | Double rail memory address signals |
| YSKRH | 150-K11 | Digit selection sink command pulse |
| YSKRL | 150-K9 | Digit selection sink command pulse |
| YSWRH | 150-K7 | Digit selection switch command pulse |
| YSWRL | 150-E4 | Digit selection switch command pulse |
| 01YD1X to 17YD8X | 153-161, -F X | Digit selection switch outputs |
| 01YB1X to 17YB4X | 153-161, -K X | Digit selection sink outputs |
| Z x x FF | 244-XX | Z-register flip-flops, bits 1 through 16 |
| ZSEC x | 270-XX | Zero sector relocation mask, bit 2 through 7 |

NOTE: x indicates a variable character position in the mnemonic.

## SECTION II
## FLOW CHARTS/INSTRUCTION ANALYSES

The instruction analyses are detailed presentations of some key signals generated within the CPU from the time an instruction operation code is recognized until the execution is complete and the next instruction's fetch cycle is initiated. The instruction fetch cycle and I-cycle are covered on pages 2-4 through 2-7. Included in an analysis are the source, destination, and operational description of the signal. The location of the logic making up the signal is also given. Within each instruction class, the definitions of Instruction Analyses are presented in alphabetical order of their mnemonic code as follows:

| CODE | INSTRUCTION |
|---|---|
| **Generic** | |
| ACA | Add C to A |
| AOA | Add ONE to A |
| CAL | Clear Left Half |
| CAR | Clear Right Half |
| CHS | Complement A Sign |
| CMA | Complement A |
| CRA | Clear A-register |
| CSA | Copy Sign and Set Sign Plus |
| ENB | Enable Program Interrupt |
| HLT | Halt |
| IAB | Interchange A and B |
| ICA | Interchange Halves |
| ICL | Interchange and Clear Left Half |
| ICR | Interchange and Clear Right Half |
| INH | Inhibit Program Interrupt |
| INK | Input Keys |
| NOP | No Operation |
| OTK | Output Keys |
| RCB | Reset C to ZERO |
| SCB | Set C to ONE |
| S-- | (Skip Group) |
| SSM | Set Sign Minus |
| SSP | Set Sign Plus |
| TCA | Two's Complement A |

| CODE | INSTRUCTION |
|---|---|
| **Input-Output** | |
| INA | Input to A |
| OCP | Output Control Pulse |
| OTA | Output from A |
| SMK | Set Mask |
| SKS | Skip if Sense Line Set |
| **Memory Reference** | |
| ADD | Add |
| ANA | Logical AND |
| CAS | Compare |
| ERA | Exclusive OR |
| IMA | Interchange memory and A |
| IRS | Increment, Replace, and Skip |
| JMP | Unconditional Jump |
| JST | Jump and Store Location |
| LDA | Load A |
| LDX | Load Index Register |
| STA | Store A |
| STX | Store Index Register |
| SUB | Subtract |
| **Shift** | |
| ALR | Logical Left Rotate |
| ALS | Arithmetic Left Shift |
| ARR | Logical Right Rotate |
| ARS | Arithmetic Right Shift |

| CODE | INSTRUCTION |
|---|---|
| Shift (Cont) | |
| LGL | Logical Left Shift |
| LGR | Logical Right Shift |
| LLL | Long Left Logical Shift |
| LLR | Long Left Rotate |
| LLS | Long Arithmetic Left Shift |
| LRL | Long Right Logical Shift |
| LRR | Long Right Rotate |
| LRS | Long Arithmetic Right Shift |

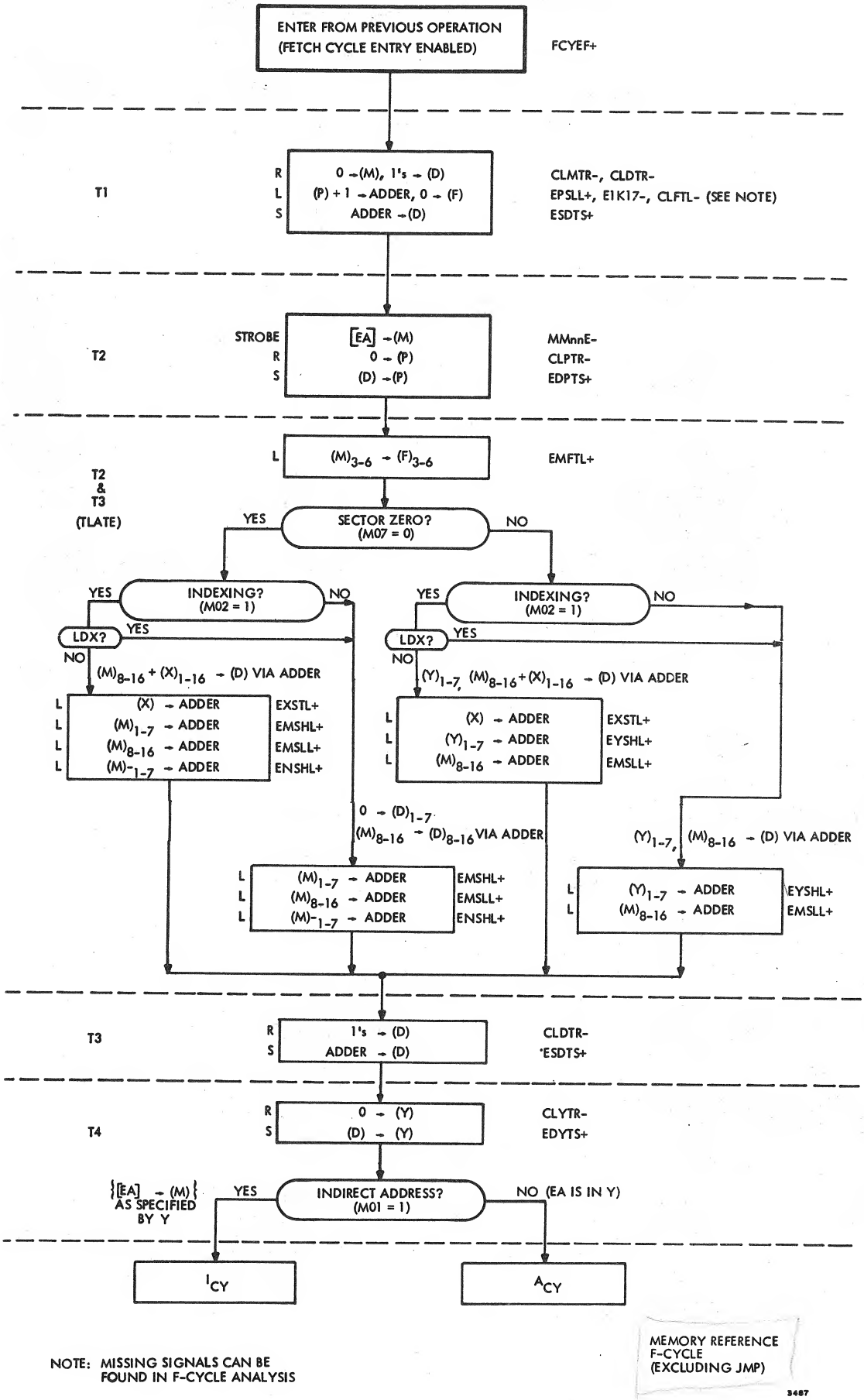
The flow charts summarize the sequence of events that occur within the central processor (CPU) during the execution of each instruction. The flow charts are easy to use if the reader applies the following general analysis.

With reference to the Clear A (CRA) flow chart, note that to the left of each box is an alphanumeric (such as T1) and columns of letter codes (such as R, L, S). The alphanumeric is the time during which operations within the horizontal dashed lines occur. The letter codes specify whether the functions in the same horizontal plane are levels (L), controlled by the reset clock MCRST (R), or controlled by the set clock MCSET (S).

The information within a box is in abbreviated text form, that is to say that certain key phrases have been replaced with symbology. An example is 0 → (M), which means that the contents of the M-register is replaced with zeros (cleared). Another method of clearing registers used in the DDP-516 is 1's → (D). Rather than replacing the contents of the D-register with zeros, the contents are replaced with ones. This method only applies to the D-, E-, and X-registers. All other registers are cleared to all zeros.

The mnemonics to the right of the box are the signals that implement the operation described in the box. In the case of 0 → (M) it is clear M-register totally on reset clock (CLMTR-) which causes the register to be cleared.

In some cases mnemonics are deliberately omitted from the flow chart for simplicity. In these cases, reference to the analyses of instructions is required.



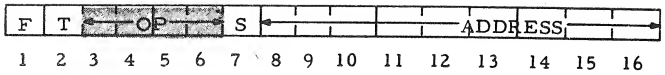

NOTE: MISSING SIGNALS CAN BE FOUND IN F-CYCLE ANALYSIS

Memory Reference F-Cycle (excluding JMP)

Instruction:

OP Code: Type:

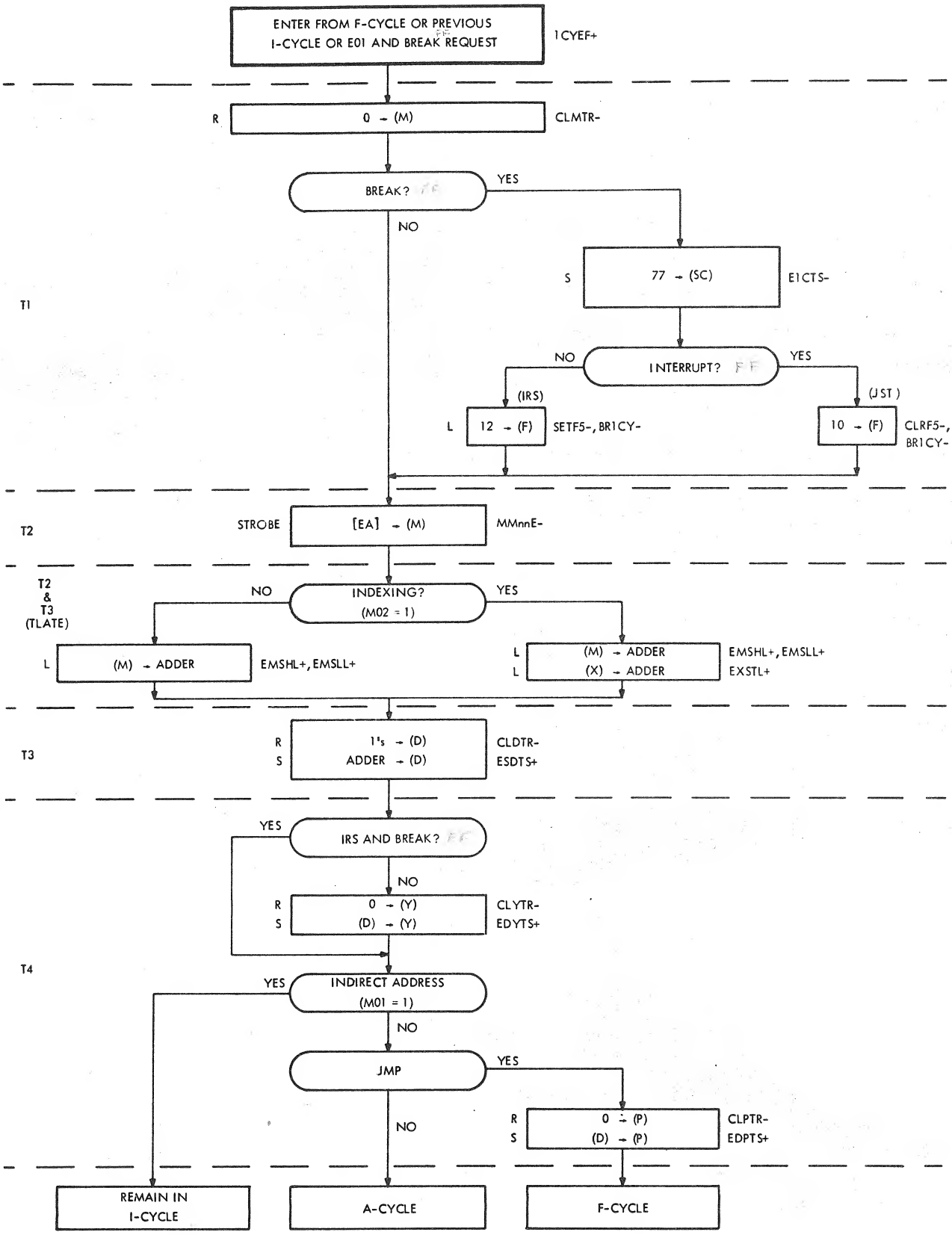
Description:




Execution Time (μsec):

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-J5 | 116-D7/D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) (MCRST+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) (MCRST+) | 125-A5 | 120-B1<br>121-A5<br>125-L10<br>124-L5<br>124-L7 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F<br>Clear MAD FF<br>Clear D0G FF |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) (MCSET+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLSEX+ | 134-K6 | F | TL1 | L | (TL1FF+)(1CYEF-) | 134-K4 | 135-B4 | Clear single execute FF |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (EDPTL+)(MCRST+) | 129-H10 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (EDPTL+)(MCSET+) | 129-H9 | 101--116-G11 | Enabled D-register into P-register |
| EMFTL+ | 128-K3 | F | TLATE | L | (M07FF-)(MEMAC-)(FCYLF+) (TLATE+)(GEN0P-)<br>or<br>(M07FF+)(MEMAC-)(FCYLF+) (TLATE+)(GEN0P-) | 127-G4<br><br>128-J2 | 120-A2<br>136-B4<br>136-D3 | Enable M(3-6) into F(3-6)<br>Bank switching control |
| EMFTL- | 128-K3 | F | TLATE | L | See EMFTL+ | 127-G4 | 127-L9 | Generate EMSLL+ |
| EXSTL+ | 128-K5 | F | TLATE | L | (F01CY+)(TLATE+)(M02FF+) (GEN0P-)(MEMAC-) (M03FF+ or M04FF+ or M05FF- or M06FF+) | 128-G6 | 101--116-A5 | Enable X-register to adder |
| EMSHL+ | 127-L8 | F | TLATE | L | (M07FF-)(MEMAC-)(FCYLF+) (TLATE+)(GEN0P-) | 127-G4 | 101--107-A8 | Enable M(1-7) to adder |
| EMSLL+ | 127-L9 | F | TLATE | L | (EMFTL-) | 128-K3 | 108--116-A8 | Enable M(8-16) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (M07FF-)(MEMAC-)(FCYLF+) (TLATE+)(GEN0P-) | 127-G4 | 101--107-A9 | Enable M-(1-7) to adder |
| EYSHL+ | 128-K3 | F | TLATE | L | (FCYLF+)(TLATE+)(GEN0P+) (MEMAC-)(M07FF+) | 128-J3 | 101--107-A11 | Enable Y-register (1-7) to adder |
| CLDTR- | 125-J5 | F | TL3 | R | (ANA0P-)(TL3FF+)(MCRST+)<br>or<br>(ACYLF-)(TL3FF+)(MCRST+) | 125-A6<br><br>125-D5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (10GRP-)(TL3FF+) | 125-D6 | 101-116-D4-D8 | Enable adder sum to D-register |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYLF-)(TL4FF+) (MCRST+) | 129-A1/D3 | 101--116-L11 | Clear Y-register |
| EDYTS+ | 129-L1 | F | TL4 | S | (ACYNX-)(MCSET+)(TL4FF+) (BRREQ-)(P1SEX+) | 129-H1 | 101--116-G10 | Enable D-register into Y-register |
| MEMCJ+ | 126 J11 | F | TL4 | L | (TL4 FF+) SPM00-) | | 150 - C1 | ENABLE SET |

(F-CYCF+) 126-F11 RCYF1+




C3688

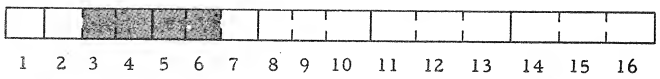
STANDARD I-CYCLE

Standard I-Cycle With Breaks

Instruction:

OP Code: Type:

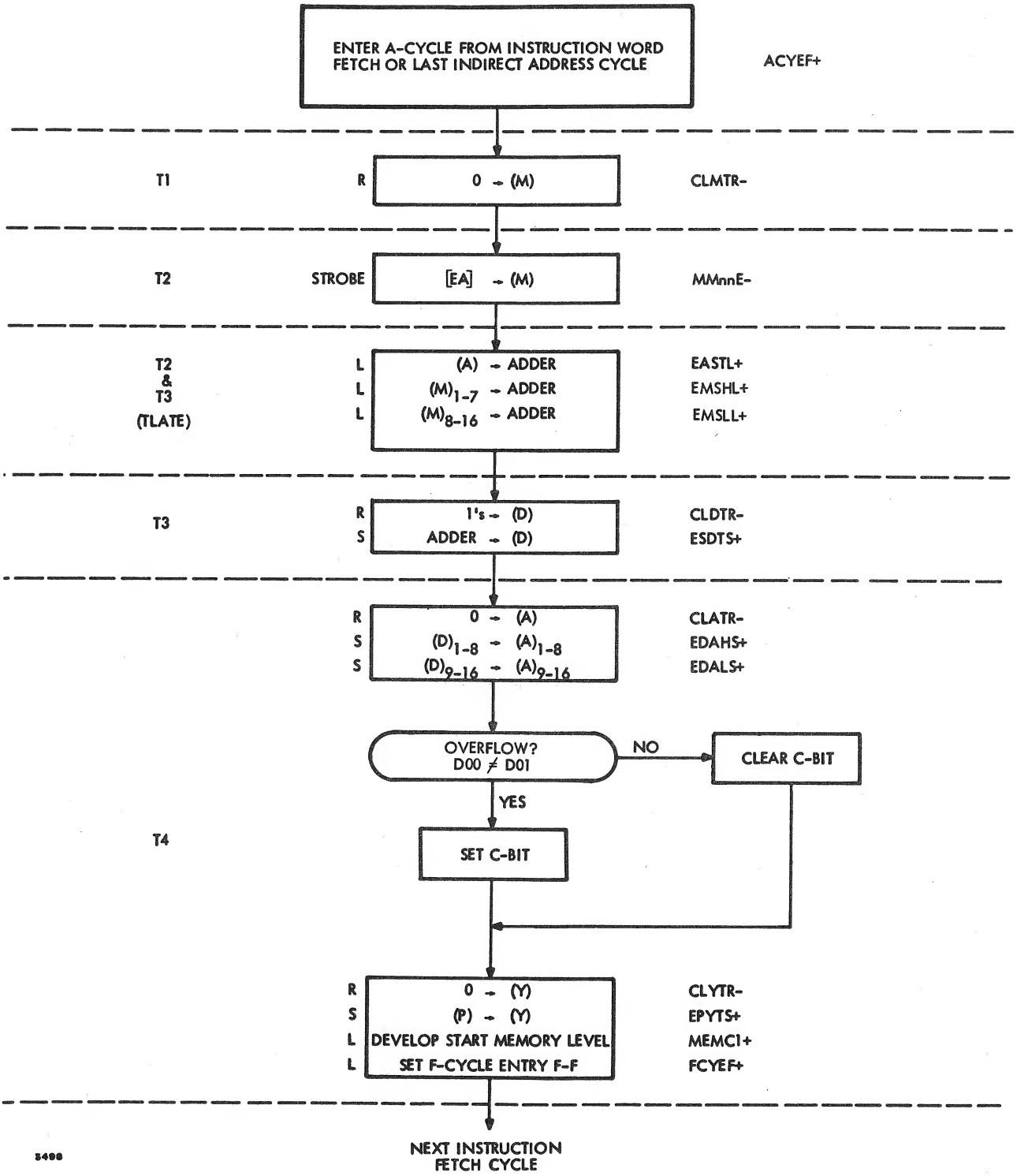
Description:




Execution Time ($\mu$sec):

| Function | Origin | Cyc | Tim | Clk | Boolean Expression | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EMSHL+ | 127-L8 | | | | (1CYEF-) | 127-L8 | 101--107-A8 | Enable M(1-7) to adder |
| EMSLL+ | 127-L10 | | | | (1CYEF-) | 127-L9 | 108--116-A9 | Enable M(8-16) to adder |
| E1K17- | 127-L4 | I | TLATE | L | (TLATE-) | 127-J6 | 116-D7/D8 | Force carry to adder |
| CLRF5+ | 134-F6 | I | TL1 | L | [(SEXRQ-)V(1G063-)]∧ [(1AD61-)(BREAK+) (1CYEF+)(TL1FF+)] | 134-C5/ E7 | 134-E4<br>120-C3 | Reset PERM1 FF<br>Clear F-register bit 5 |
| SETF5- | 134-H5 | I | TL1 | L | [(SEXRQ+)(1G063+)V (1AD61+)]∧[(BREAK+) (1CYEF+)(TL1FF+)] | 134-X5 | 120-C4 | Set F-register bit 5 |
| BR1CY- | 134-C6 | I | TL1 | L | (BREAK+)(1CYEF+)(TL1FF+) | 134-C6 | 120-B1-B8 | Force F-register to JST0P- or 1RS0P- depending on F05 |
| CLMTR- | 128-K8 | I | TL1 | R | (MCRST+)(ACYEF-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| E1CTS- | 125-J8 | I | TL1 | S | (BR1CY+)(MCSET+)(AZZZZ-) | 125-H8 | 126-F5<br>126-D8<br>121-X8 | Generate RPTT2+ (Don't care)<br>Set WR1NH FF<br>Set shift counter to $77_8$ |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| EXSTL+ | 128-K5 | I | TLATE | L | (1CYLF+)(F01CY+)(TLATE+) (M02FF+)(GEN0P-)(MEMAC-) | 128-J5 | 101--116-A6 | Enable X-register to adder |
| CLDTR- | 125-J5 | I | TL3 | R | (ACYLF-)(TL3FF+)(MCRST+) | 125-D5 | 101--116-F7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | I | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-D6 | 101--116-D5-D9 | Enable adder sum to D-register |
| MEMC1+ | 126-H11 | I | TL4 | L | (10GRP-)(SPM0D-)(TL4FF+) | 126-F11 | 150-D1 | Enable set RCYF1+ |
| CLPTR- | 129-J10 | I | TL4 | R | (0PGJS+)(E01NS+)(TL4FF+) (MCRST+) | 129-D9 | 101--116-H10 | Clear P-register |
| CLYTR- | 129-J3 | I | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-D3/ H3 | 101--116-L11 | Clear Y-register* |
| EDPTS+ | 129-L9 | I | TL4 | S | (0PGJS+)(E01NS+)(TL4FF+) (MCSET+) | 129-D9 | 101--116-G11 | Enable D-register into P-register |
| EDYTS+ | 129-L1 | I | TL4 | S | [(MCSET+)(TL4FF+) (ACYNX-)]∧[(BRREQ-)V (DCYXX-)(E01NS-)] | 129-H1/ H2 | 101--116-G10 | Enable D-register into Y-register* |
| E1YLS+ | 129-L6 | I | TL4 | S | (TL4FF+)(BRREQ+)∧ [(DCYXX-)(E01NS-)] -∧ (MCSET+) | 125-D5/ H6 | 110--116-K10 | Enable 1YB(10-16) into Y(10-16) |

*Y-register changed except when [(F01CY+)(BREAK+)(1RS0P+)] is TRUE

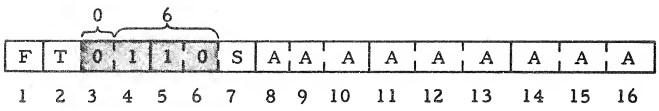



ADD
2 CYCLES
OP CODE 06

Instruction: Add (ADD)

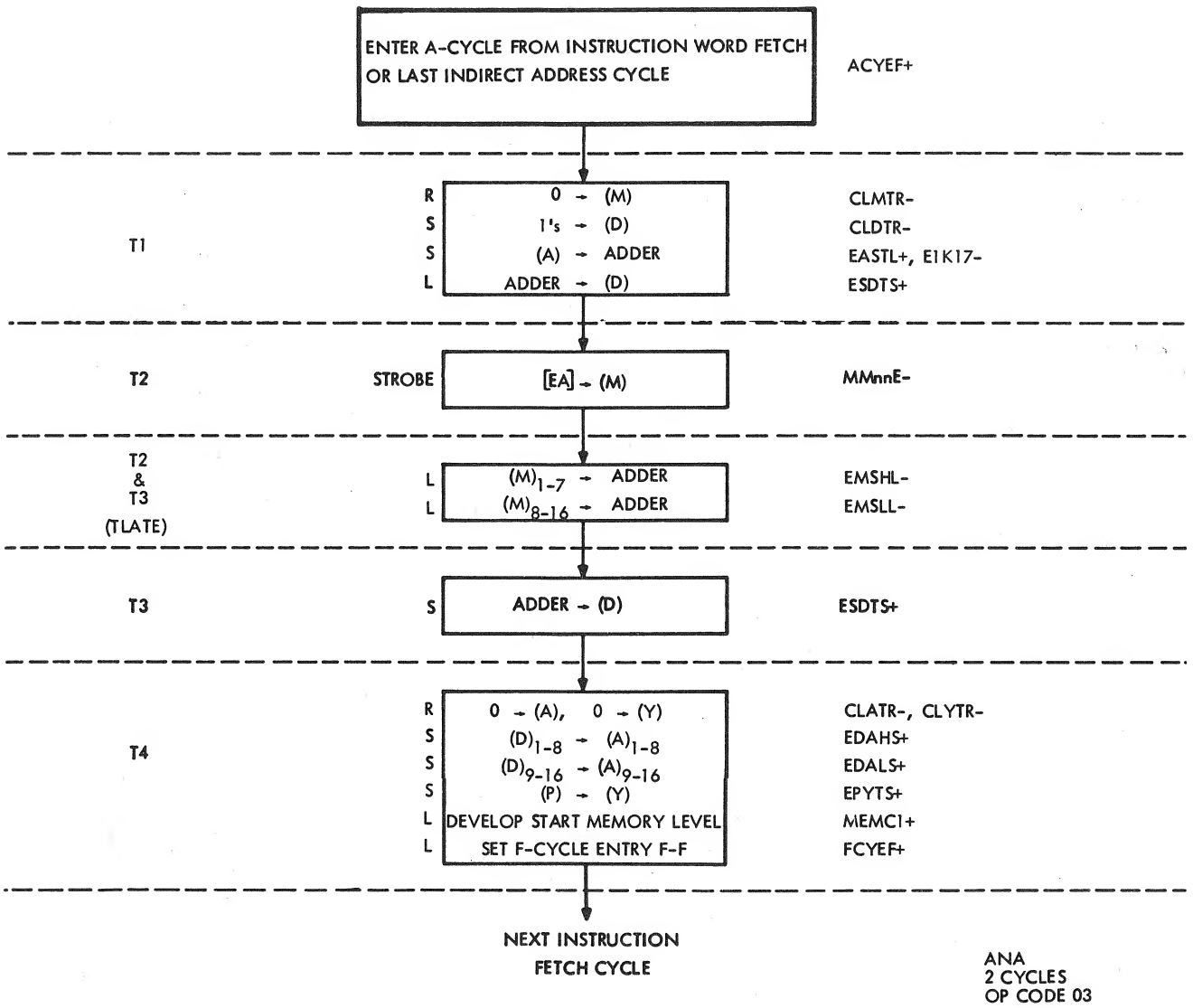
OP Code: 06 Type: MR, 2 cycles

Description: (A) + [EA] → A
OVF → (C)

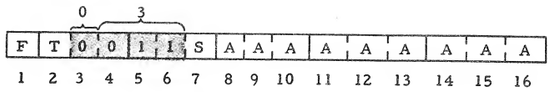



Execution Time (μsec): 1.92

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| EASTL+ | 127-L1 | A | TLATE | L | (ADD0P+)(ACYLF+) (TLATE+) | 127-A1/ C1 | 101--116-A5 | Enable A-register to adder |
| EMSHL+ | 127-L8 | A | TLATE | L | (ACYLF+)(TLATE+)(SUB0P-) (0PGAA+) | 127-G8 | 101--107-A8 | Enable M-register (1-7) to adder |
| EMSLL+ | 127-L10 | A | TLATE | L | (ACYLF+)(TLATE+)(SUB0P-) (0PGAA+) | 127-G8 | 108--116-A8 | Enable M(8-16) to adder |
| CLDTR- | 125-J5 | A | TL3 | R | (ANA0P-)(TL3FF+) | 125-B6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | A | TL3 | S | (10GRP-)(TL3FF+) | 125-D7 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLATR- | 122-H7 | A | TL4 | R | (ACYLF+)(0PGAA+)(TL4FF+) | 122-D3 | 101--116-H5 | Clear A-register |
| EDAHS+ | 122-L1 | A | TL4 | S | (ACYLF+)(0PGAA+)(TL4FF+) | 122-D3 | 101--108-G7 | Enable D-register to A(1-8) |
| EDALS+ | 122-L2 | A | TL4 | S | (ACYLF+)(0PGAA+)(TL4FF+) | 122-D3 | 109--116-G7 | Enable D-register to A(9-16) |
| CB1TF+ | 124-L2 | A | TL4 | L | (ADD0P+)(D00 = D01) | 124-A4 | 124-L2 | Set CB1TF |
| CB1TF- | 124-L1 | A | TL4 | L | (ADD0P+)(D00 ≠ D01) | 124-A5 | 124-L1 | Reset CB1TF |
| CLYTR- | 129-J3 | A | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-A1 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | A | TL4 | S | (P1SEX-)(E01NS+)(0PGJS-) | 129-D4/ H4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/ H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

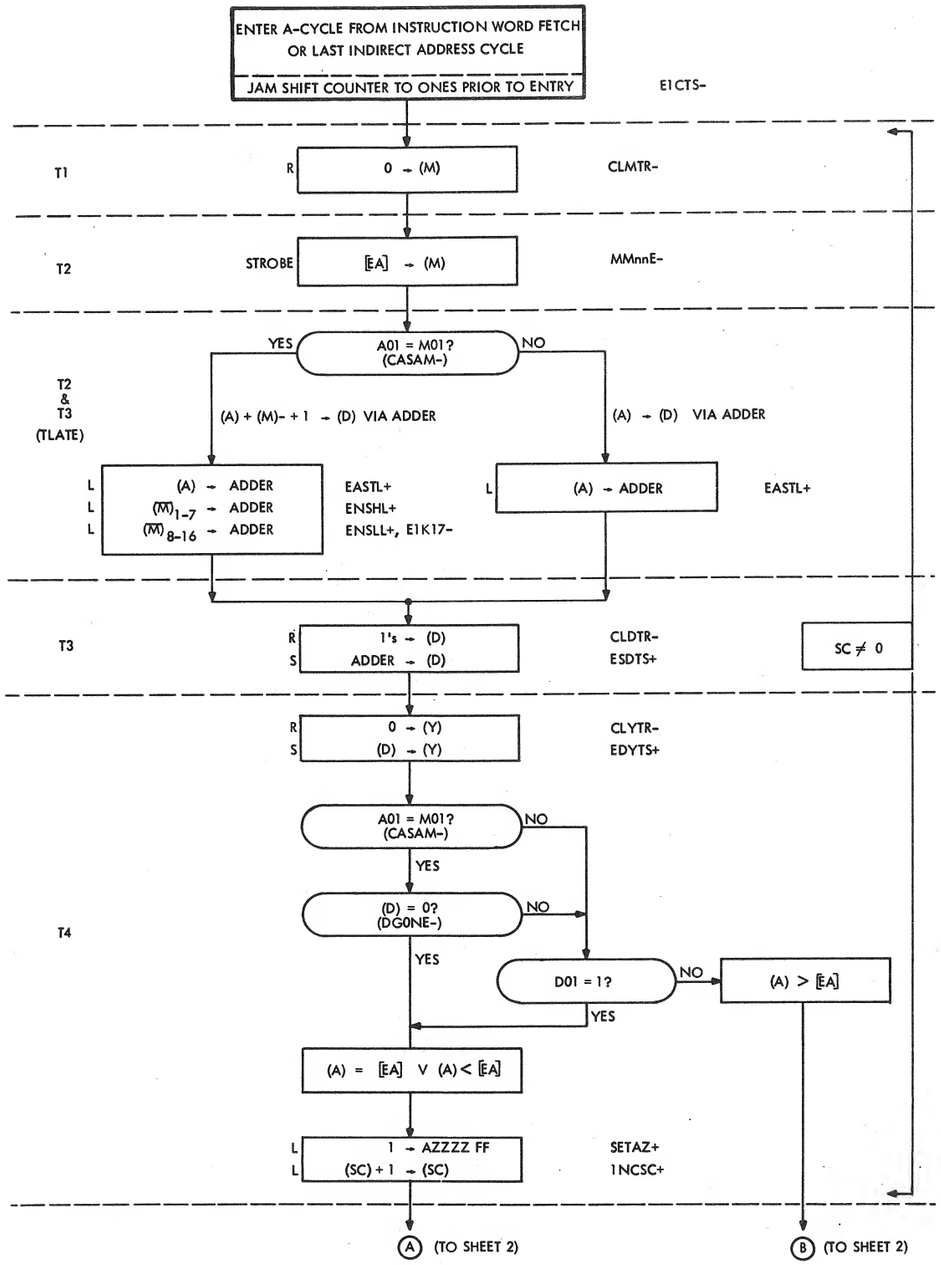

ENTER A-CYCLE FROM INSTRUCTION WORD FETCH OR LAST INDIRECT ADDRESS CYCLE
ACYEF+
T1
R 0 → (M) CLMTR-
S 1's → (D) CLDTR-
S (A) → ADDER EASTL+, E1K17-
L ADDER → (D) ESDTS+
T2
STROBE
[EA] → (M)
MMnnE-
T2 & T3 (TLATE)
L (M)1-7 → ADDER EMSHL-
L (M)8-16 → ADDER EMSLL-
T3
S ADDER → (D) ESDTS+
T4
R 0 → (A), 0 → (Y) CLATR-, CLYTR-
S (D)1-8 → (A)1-8 EDAHS+
S (D)9-16 → (A)9-16 EDALS+
S (P) → (Y) EPYTS+
L DEVELOP START MEMORY LEVEL MEMC1+
L SET F-CYCLE ENTRY F-F FCYEF+
NEXT INSTRUCTION FETCH CYCLE
ANA
2 CYCLES
OP CODE 03
3803

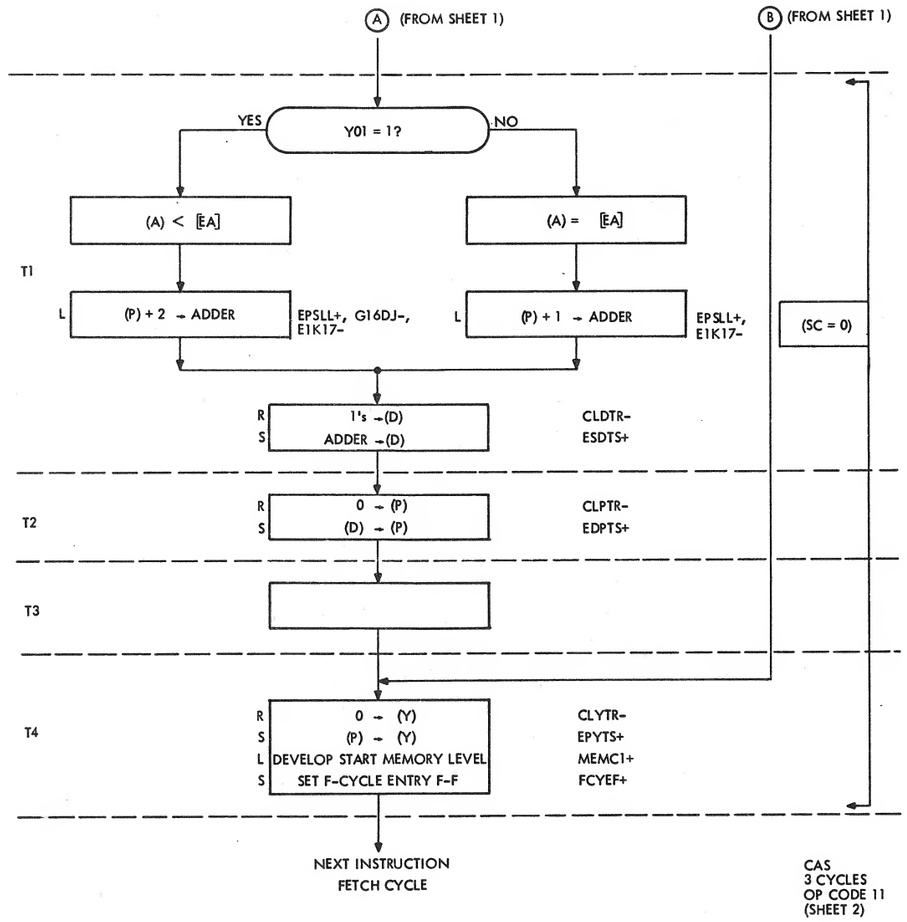
Instruction: Logical AND (ANA)
OP Code: 03 Type: MR, 2 cycles
Description: (A) ∧ [EA] → (A)




Execution Time (μsec): 1.92

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| E1K17+ | 127-J6 | A | TL1 | L | (TLATE-) | 127-J6 | 116-D7/D9 | Force carry-in to bit 16 |
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | A | TL1 | R | (ACYEF+)(TL1FF+)(JST0P-)(1RS0P-)(1MA0P-) | 125-D4 | 101--116-E7 | Clear D-register to ONEs |
| EASTL+ | 127-L1 | A | TL1 | L | (ACYEF+)(TLATE-)(CAS0P-)(LSX0P-)(10GRP-) | 127-G10 | 101--116-A5 | Enable A-register to adder |
| ESDTS+ | 125-L4 | A | TL1 | S | (ACYEF+)(TL1FF+)(JST0P-)(1RS0P-)(1MA0P-) | 125-D4 | 101--116-D4/D6/D8 | Enable adder to D-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| EMSHL+ | 127-L8 | A | TLATE | L | (ACYLF+)(TLATE+)(SUB0P-)(0PGAA+) | 127-G8 | 101--107-A8 | Enable M(1-7) to adder |
| EMSLL+ | 127-L10 | A | TLATE | L | (ACYLF+)(TLATE+)(SUB0P-)(0PGAA+) | 127-G8 | 108--116-A8 | Enable M(8-16) to adder |
| ESDTS+ | 125-L4 | A | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-D7/H4 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLATR- | 122-H7 | A | TL4 | R | (ACYLF+)(TL4FF+)(0PGAA+)(1MA0P-) | 122-D2 | 101--116-H5 | Clear A-register |
| EDAHS+ | 122-L1 | A | TL4 | S | (ACYLF+)(TL4FF+)(0PGAA+)(1MA0P-) | 122-D2 | 101--108-G7 | Enable D-register to A(1-8) |
| EDALS+ | 122-L2 | A | TL4 | S | (ACYLF+)(TL4FF+)(0PGAA+)(1MA0P-) | 122-D2 | 109--116-G7 | Enable D-register to A(9-16) |
| CLYTR- | 129-J3 | A | TL4 | R | (SCZR0-) | 129-A1 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | A | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

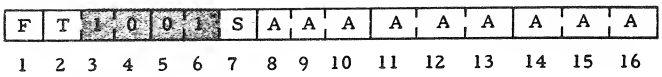

ENTER A-CYCLE FROM INSTRUCTION WORD FETCH OR LAST INDIRECT ADDRESS CYCLE
JAM SHIFT COUNTER TO ONES PRIOR TO ENTRY
E1CTS-
T1
R
0 → (M)
CLMTR-
T2
STROBE
[EA] → (M)
MMnnE-
YES
A01 = M01? (CASAM-)
NO
T2 & T3 (TLATE)
(A) + (M)- + 1 → (D) VIA ADDER
(A) → (D) VIA ADDER
L
(A) → ADDER
(M)1-7 → ADDER
(M)8-16 → ADDER
EASTL+
ENSHL+
ENSLL+, E1K17-
L
(A) → ADDER
EASTL+
T3
R
S
1's → (D)
ADDER → (D)
CLDTR-
ESDTS+
SC ≠ 0
R
S
0 → (Y)
(D) → (Y)
CLYTR-
EDYTS+
A01 = M01? (CASAM-)
NO
YES
(D) = 0? (DG0NE-)
NO
YES
T4
D01 = 1?
NO
YES
(A) > [EA]
(A) = [EA] V (A) < [EA]
L
1 → AZZZZ FF
L
(SC) + 1 → (SC)
SETAZ+
1NCSC+
A (TO SHEET 2)
B (TO SHEET 2)
CAS
3 CYCLES
OP CODE 11
(SHEET 1)
3813


A (FROM SHEET 1)
B (FROM SHEET 1)
YES
Y01 = 1?
NO
T1
(A) < [EA]
(A) = [EA]
L
(P) + 2 → ADDER
EPSLL+, G16DJ-, E1K17-
L
(P) + 1 → ADDER
EPSLL+, E1K17-
(SC = 0)
R
S
1's → (D)
ADDER → (D)
CLDTR-
ESDTS+
T2
R
S
0 → (P)
(D) → (P)
CLPTR-
EDPTS+
T3
T4
R
S
L
S
0 → (Y)
(P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY F-F
CLYTR-
EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION FETCH CYCLE
CAS
3 CYCLES
OP CODE 11
(SHEET 2)
3813

Instruction: Compare (CAS)

OP Code: 11 Type:

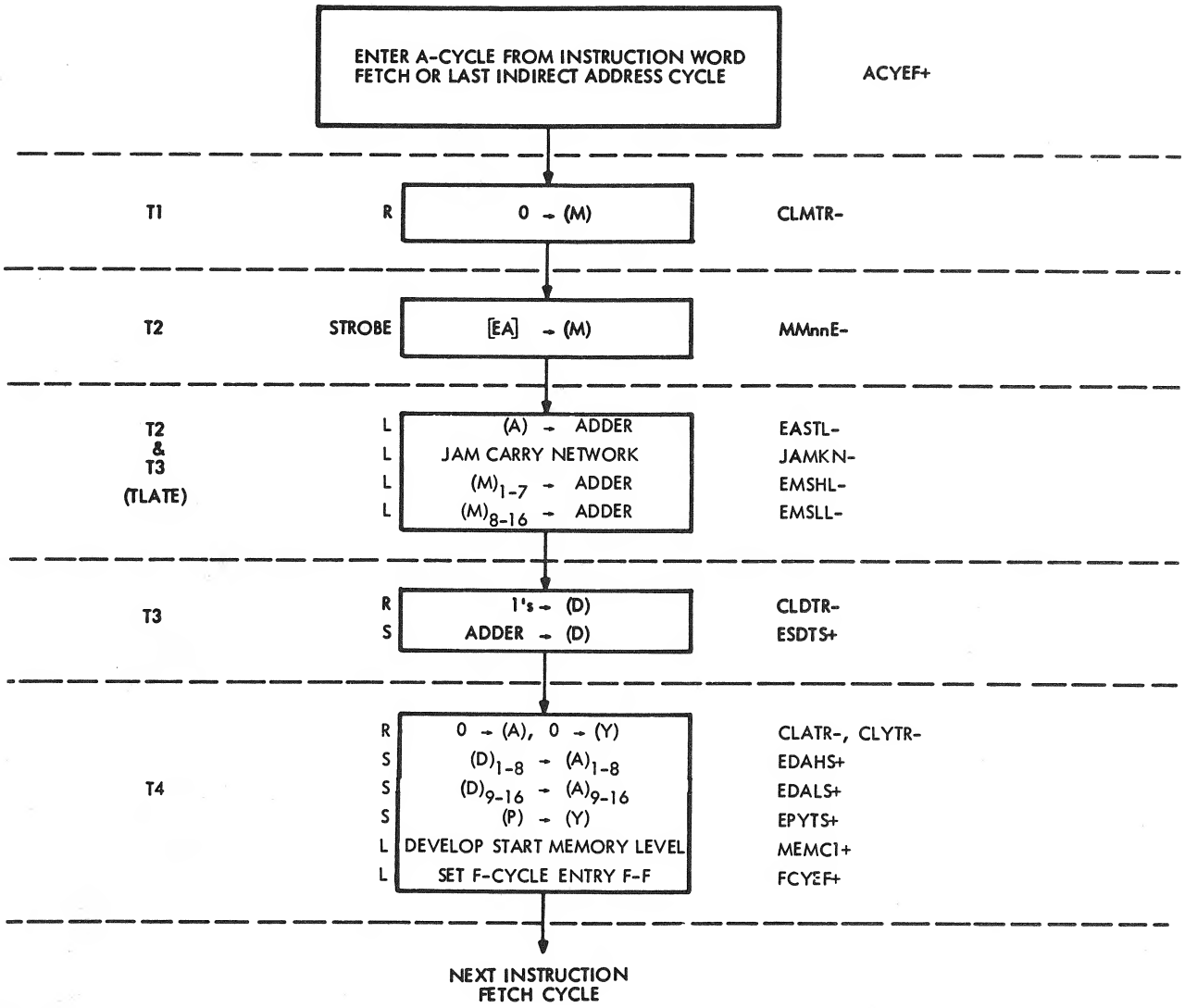
Description: If: (A) > [EA] = No operation
(A) = [EA] = Skip next instruction
(A) < [EA] = Skip two instructions




Execution Time (μsec): 2.88

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| CASAM- | 126-L4 | | | | (CAS0P+)∧[(A01FF-)(M01FF-) V(A01FF+)(M01FF+)] | 126-L4 | 125-B11 | M01 = A01 |
| ACYEF+ | 119-F4 | F | TL4 | L | (TL4FF+)(E01NS-)(F01CY+) [(M01FF-) at 119-B5] | 119-D5 | 119-H3 | Set A-cycle |
| E1CTS- | 125-J8 | F | TL4 | S | (FCYLF+)(TL4FF+)(0PG3C+) | 125-A7 | 121-A8 | ONEs to shift counter |
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| EASTL+ | 127-L1 | A | TLATE | L | (ACYLF+)(TLATE+)(CAS0P-) | 127-C1 | 101--116-A5 | Enable A-register to adder |
| ENSHL+ | 127-L7 | A | TLATE | L | (ACYLF+)(TLATE+)(0PGNS+) (1RS0P-) | 127-C11 | 101--107-A9 | Enable M-(1-7) to adder |
| ENSLL+ | 127-L5 | A | TLATE | L | (ACYLF+)(TLATE+)(0PGNS+) (1RS0P-) | 127-C11 | 108--116-A9 | Enable M-(8-16) to adder |
| E1K17- | 127-L4 | A | TLATE | L | (CAS0P+)(ACYLF+) | 127-E7/ J5 | 116-D7-D9 | Force carry to adder |
| | | | | | | | | SC ≠ 0 |
| CLDTR- | 125-J5 | A | TL3 | R | (ANA0P-)(TL3FF+)(MCRST+) | 125-B6 | 101--116-F7 | ONEs to D-register |
| ESDTS+ | 125-L4 | A | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLYTR- | 129-J3 | A | TL4 | R | (CAS0P+)(TL4FF+)(MCRST+) (MCRST+) | 129-A1/ D3 | 101--116-L11 | Clear Y-register |
| EDYTS+ | 129-L1 | A | TL4 | S | (BRREQ-)(E01NS-)(CAS0P+) (MCSET+)(TL4FF+) | 129-B1/ D4/H1 | 101--116-G10 | Enable D-register to Y-register |
| DG0NE- | 126-B9 | | | | (D01FF-) through (D16FF-) | 126-B9 | 125-D11 | D-register equals ZERO |
| SETAZ+ | 125-J9 | A | TL4 | L | (ACYLF+)(TL4FF+)(CAS0P+) ∧(DG0NE+)∨(D01FF+) | 125-D11∨ 125-B11 | 125-L10 | Set AZZZZ F-F |
| INCSC+ | 126-L3 | A | TL4 | L | (ACYLF+)(TL4FF+) * | 126-H4 | 121-A5 | Increment shift counter |
| G16DJ- | 116-A2 | A | TL1 | L | (TLATE-)(ACYEF+)(CAS0P+) (Y01FF+) | 116-A2 | 116-D7 117-B5 | If (A) < [EA] force additional carry to adder |
| EPSLL+ | 128-H4 | A | TL1 | L | (CAS0P+)(TLATE-) | 128-D4 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | A | TLATE | L | (TLATE-) | 127-J6 | 116-D7-D9 | Force carry to adder |
| CLDTR- | 125-D4 | A | TL1 | R | (ACYEF+)(TL1FF+)(JST0P-) (1RS0P-)(1MA0P-)(MCRST+) | 125-D4 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | A | TL1 | S | (ACYEF+)(TL1FF+)(JST0P-) (1RS0P-)(1MA0P-)(MCSET+) | 125-L4 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLPTR- | 129-J10 | A | TL2 | R | (ACYEF+)(TL2FF+)(CAS0P+) (AZZZZ+)(MCRST+) | 129-D10 | 101--116-H10 | Clear P-register |
| | | | | | | | | SC = 0 |
| EDPTS+ | 129-L9 | A | TL2 | S | (ACYEF+)(TL2FF+)(CAS0P+) (AZZZZ+)(MCSET+) | 129-D10 | 101--116-G11 | Enable D-register to P-register |
| CLYTR- | 129-J3 | A | TL4 | R | (CAS0P+)(TL4FF+)(MCRST+) | 129-A1/ D1 | 101--116-L11 | Clear Y-register |
| EPYTS+ | 129-L4 | A | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+) (0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/ H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

*(SC) = 1 from previous F-cycle)



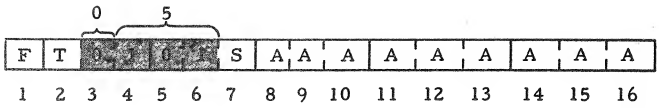

3811

ERA
2 CYCLE
OP CODE 05

Instruction: Exclusive OR (ERA)

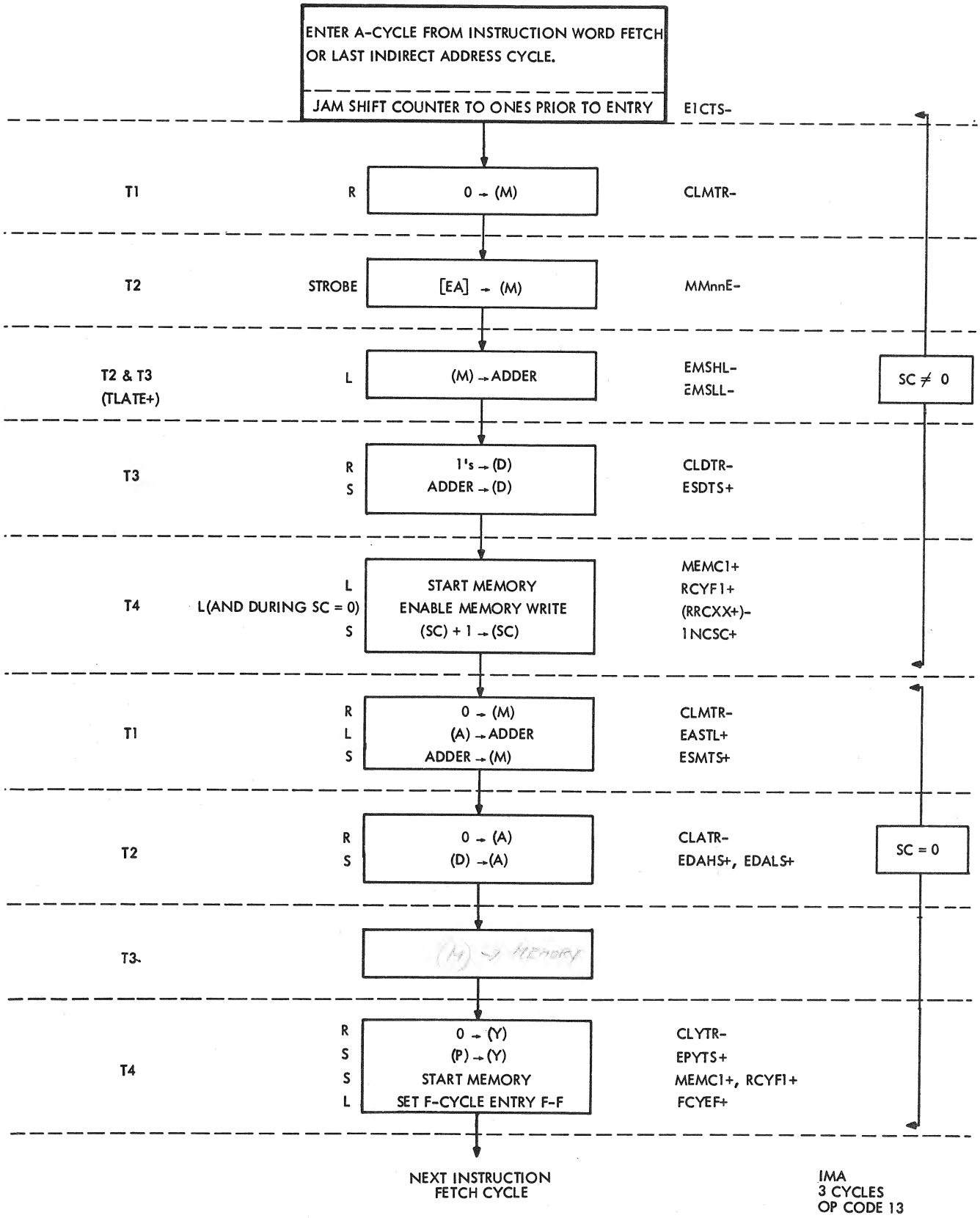
OP Code: 05 Type: MR, 2 cycle

Description: (A)$\forall$[EA] → (A)




Execution Time ($\mu$sec): 1.92

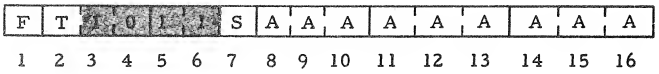
| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| EASTL+ | 127-L1 | A | TLATE | L | (ERA0P+)(TLATE+)(ACYLF+) | 127-B1 | 101--116-A5 | Enable A-register to adder |
| JAMKN- | 127-J3 | A | TLATE | L | (ACYEF+)(ERA0P+) | 127-C3 | 101--117-C8 | Jam carry network |
| EMSHL+ | 127-L8 | A | TLATE | L | (ACYLF+)(TLATE+)(SUB0P-)(0PGAA+) | 127-G8 | 101--107-A8 | Enable M(1-7) to adder |
| EMSLL+ | 127-L10 | A | TLATE | L | (ACYLF+)(TLATE+)(SUB0P-)(0PGAA+) | 127-G8 | 108--116-A8 | Enable M(8-16) to adder |
| CLDTR- | 125-J5 | A | TL3 | R | (ANA0P-)(TL3FF+)(MCRST+) | 125-A6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | A | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLATR- | 122-H7 | A | TL4 | R | (ACYLF+)(TL4FF+)(0PGAA+)(1MA0P-) | 122-D2 | 101--116-H5 | Clear A-register |
| EDAHS+ | 122-L1 | A | TL4 | S | (ACYLF+)(TL4FF+)(0PGAA+)(1MA0P-) | 122-D2 | 101--108-G7 | Enable D-register to A-register (1-8) |
| EDALS+ | 122-L2 | A | TL4 | S | (ACYLF+)(TL4FF+)(0PGAA+)(1MA0P-) | 122-D2 | 109--116-G7 | Enable D-register to A(9-16) |
| CLYTR- | 129-J3 | A | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-A1 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | A | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |


ENTER A-CYCLE FROM INSTRUCTION WORD FETCH OR LAST INDIRECT ADDRESS CYCLE.
JAM SHIFT COUNTER TO ONES PRIOR TO ENTRY
E1CTS-
T1
R
0 → (M)
CLMTR-
T2
STROBE
[EA] → (M)
MMnnE-
T2 & T3 (TLATE+)
L
(M) → ADDER
EMSHL-
EMSLL-
SC ≠ 0
T3
R
S
1's → (D)
ADDER → (D)
CLDTR-
ESDTS+
T4
L (AND DURING SC = 0)
L
S
START MEMORY
ENABLE MEMORY WRITE
(SC) + 1 → (SC)
MEMC1+
RCYF1+
(RRCXX+)-
1NCSC+
T1
R
L
S
0 → (M)
(A) → ADDER
ADDER → (M)
CLMTR-
EASTL+
ESMTS+
T2
R
S
0 → (A)
(D) → (A)
CLATR-
EDAHS+, EDALS+
SC = 0
T3.
(M) → MEMORY
T4
R
S
S
L
0 → (Y)
(P) → (Y)
START MEMORY
SET F-CYCLE ENTRY F-F
CLYTR-
EPYTS+
MEMC1+, RCYF1+
FCYEF+
NEXT INSTRUCTION FETCH CYCLE
IMA
3 CYCLES
OP CODE 13
3302

Instruction: Interchange memory and A (IMA)

OP Code: 13 Type: MR, 3 cycle

Description: (EA) ⇄ (A)

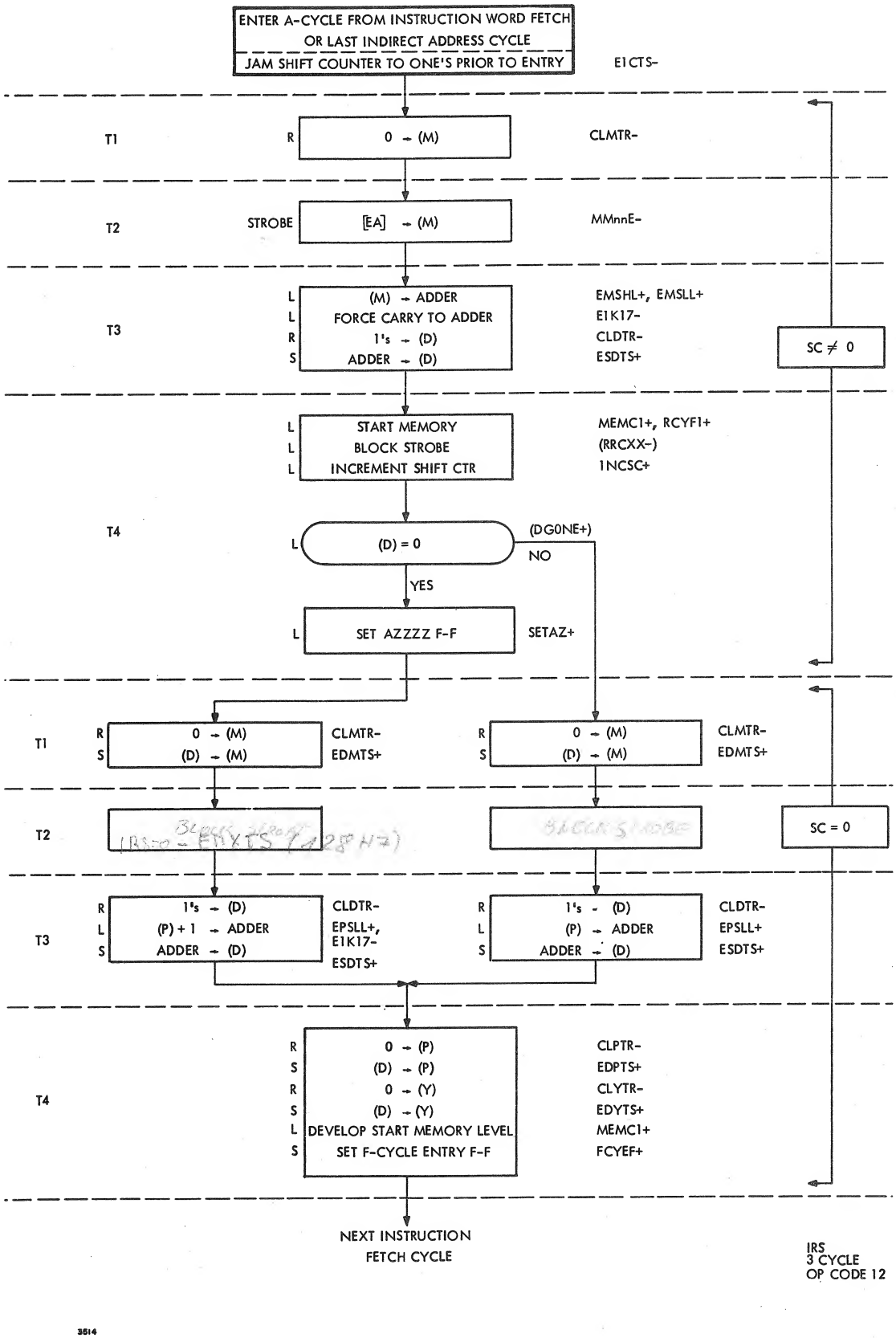



Execution Time ($\mu$sec): 2.88

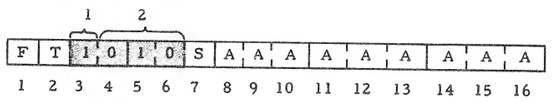
| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| ACYEF+ | 119-F4 | F | TL4 | L | (TL4FF+)(E01NS-)(F01CY+)[(M01FF-) @ 119-B5] | 119-D5 | 119-H3 | Set A-cycle at next TL1 |
| E1CTS- | 125-J8 | F | TL4 | S | (FCYLF+)(TL4FF+)(0PG3C+)* | 125-A7 | 121-A8 | Jam shift counter to ONEs |
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Reset M-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| EMSHL+ | 127-L8 | A | TLATE | | [(ACYLF+)(TLATE+)(SUB0P-) | 127-G8 | 101--108-A8 | Enable M(1-8) to adder |
| EMSLL+ | 127-L10 | A | TLATE | | (0PGAA+)] | 127-G8 | 109--116-A8 | Enable M(9-16) to adder |
| CLDTR- | 125-J5 | A | TL3 | R | (ANA0P-)(TL3FF+) | 125-A6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | A | TL3 | S | (TL3FF+)(10GRP-) | 125-D6 | 101--116-D4/D6/D8 | Enable adder sum to D-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (MEMC1-) | 126-G11 | 150-C1 | Enable set RCYF1+ |
| 1NCSC+ | 126-L3 | A | TL4 | L | (ACYLF+)(TL4FF+) | 126-H4 | 121-A5 | Enable step shift counter |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | XXXX | Start memory cycle |
| [A1A24-D] | 121-B5 | A | TL4 | S | (1NCSC+)(SCZR0-)(MCSET+) | 121-A5 | 121-X6 | Step shift counter to ZERO |
| RRCXX- | 126-L6 | A* | TL4 | L- | (ACYEF+)(0PGWR+)(WR1NH-) | 126-F6 | 150-D6 | Block STRB1+ to enable memory write cycle |
| | | As* | TL3 | L | | | | |
| CLMTR- | 128-K8 | As | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Reset M-register |
| EASTL+ | 127-L1 | As | TL1 | L | (ACYEF+)(TLATE-)(CAS0P-)(LSX0P-)(10GRP-) | 127-J1 | 101--116-A5 | Enable A-register to adder |
| ESMTS+ | 128-H10 | As | TL1 | S | (RRCXX-)(MAST0-)(0PGSM+)(TL1FF+)(MCSET+) | 128-F10 | 101--116-G9 | Enable adder RnnPA+ output into M-register |
| CLATR- | 122-J7 | As | TL2 | R | (CLATL+)(MCRST+)* | 122-H7 | 101--116-H5 | Reset A-register |
| EDAHS+ | 122-L1 | As | TL2 | S | [(E01NS+)(TL2FF+)(0PGAA+) | 122-A4 | 101--108-G7 | Enable D(1-8) into A(1-8) |
| EDALS+ | 122-L2 | As | TL2 | S | (0PGSM+)] | 122-AH | 109--116-G7 | Enable D(9-16) into A(9-16) |
| CLYTR- | 129-J3 | As | TL4 | R | (TL4FF+)(ACYNX-)* | 129-D3 | 101--116-L11 | Reset Y-register |
| FCYEF+ | 119-F10 | As | TL4 | L | (Set: (TL4FF+)(E01NS+)) | 119-D10 | 119-H9 | Enable set FCYLF+ at next TL1FF+ |
| EPYTS+ | 129-L5 | As | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-) | 129-D4 | 101--116-J11 | Enable P-register into Y-register |
| MEMC1+ | 126-J11 | As | TL4 | L | (MEMC1-) | 126-G11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | As | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | XXXX | Start memory cycle |

*0PG3C+ @ 120-F9 = (1MA0P-), etc.
*A/TL4/L - As/TL3/L ≡ (ACYLF+)(TL4FF+)(SCZR0-)V(ACYEF+)(SCZR0+)
*A/ = A-cycle + SC ≠ 0, As/ = A-cycle + SC = 0

*(CLATL+) = (E01NS+)(TL2FF+)(0PGAA+)(0PGSM+) @ 122-A4
*(ACYNX-) = ((ACYLF+)(LSX0P-)(CAS0P-)(SCZR0-))-@ 129-A1

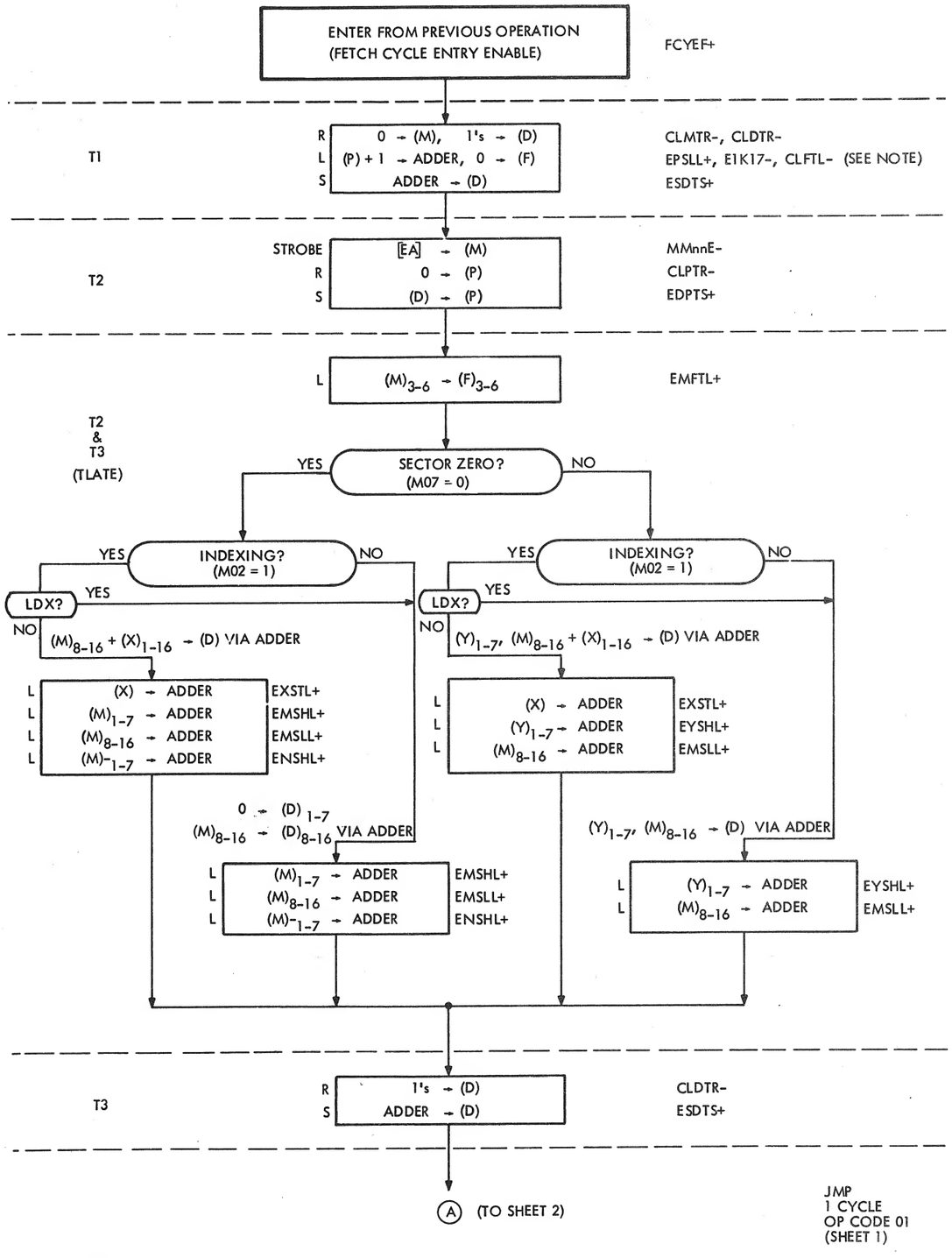

ENTER A-CYCLE FROM INSTRUCTION WORD FETCH
OR LAST INDIRECT ADDRESS CYCLE
JAM SHIFT COUNTER TO ONE'S PRIOR TO ENTRY
E1CTS-
T1
R
0 → (M)
CLMTR-
T2
STROBE
[EA] → (M)
MMnnE-
T3
L
(M) → ADDER
EMSHL+, EMSLL+
L
FORCE CARRY TO ADDER
E1K17-
R
1's → (D)
CLDTR-
S
ADDER → (D)
ESDTS+
SC ≠ 0
L
START MEMORY
MEMC1+, RCYF1+
L
BLOCK STROBE
(RRCXX-)
L
INCREMENT SHIFT CTR
1NCSC+
T4
L
(D) = 0
(DG0NE+)
NO
YES
L
SET AZZZZ F-F
SETAZ+
T1
R
0 → (M)
CLMTR-
S
(D) → (M)
EDMTS+
R
0 → (M)
CLMTR-
S
(D) → (M)
EDMTS+
T2
SC = 0
T3
R
1's → (D)
CLDTR-
L
(P) + 1 → ADDER
EPSLL+, E1K17-
S
ADDER → (D)
ESDTS+
R
1's → (D)
CLDTR-
L
(P) → ADDER
EPSLL+
S
ADDER → (D)
ESDTS+
T4
R
0 → (P)
CLPTR-
S
(D) → (P)
EDPTS+
R
0 → (Y)
CLYTR-
S
(D) → (Y)
EDYTS+
L
DEVELOP START MEMORY LEVEL
MEMC1+
S
SET F-CYCLE ENTRY F-F
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
IRS
3 CYCLE
OP CODE 12
3514

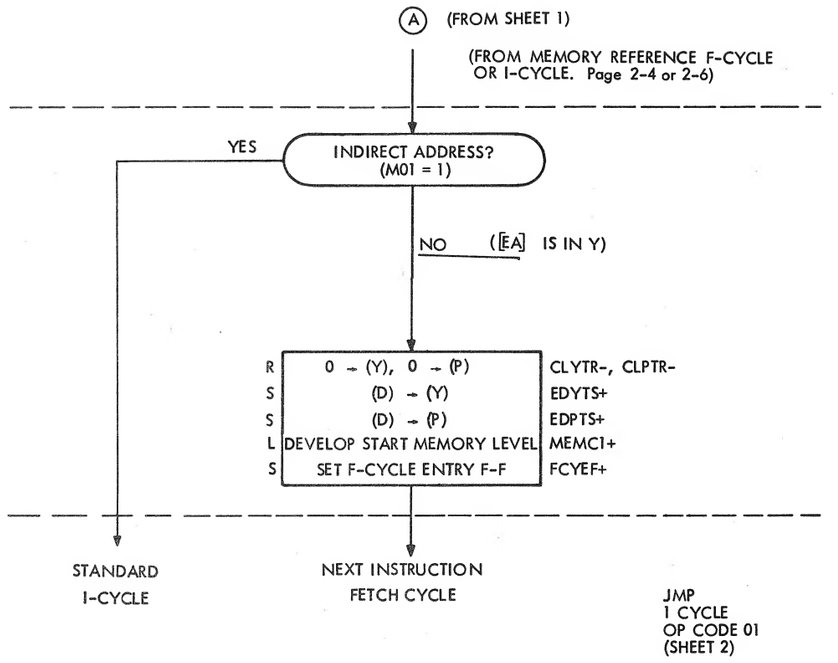
Instruction: Increment, Replace, and Skip (IRS)
OP Code: 12 Type: MR, 3 cycles
Description: [EA] + 1 → [EA]




Execution Time (μsec): 2.88

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| ACYEF+ | 119-F4 | F | TL4 | L | (TL4FF+)(E01NS-)(M01ML-)(FCYLF+) | 119-C5 | 119-H3 | Set A-cycle F-F at next TL1 |
| E1CTS- | 125-J8 | F | TL4 | S | (FCYLF+)(TL4FF+)(0PG3C+)(MCSET+)(AZZZZ-) | 125-A7/H9 | 121-A8 | Jam shift counter to ONEs |
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| EMSHL+ | 127-L8 | A | TL3 | L | (ACYEF+)(1RS0P+)(SCZR0-) | 127-E9 | 101--107-A8 | Enable M(1-7) to adder |
| EMSLL+ | 127-L10 | A | TL3 | L | (ACYEF+)(1RS0P+)(SCZR0-) | 127-E9 | 108--116-A8 | Enable M(8-16) to adder |
| E1K17- | 127-L4 | A | TL3 | L | (ACYEF+)(1RS0P+)(SCZR0-) | 127-L4 | 116-D7-D9 | Force carry to adder |
| CLDTR- | 125-J5 | A | TL3 | R | (ANA0P-)(TL3FF+) | 125-A6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | A | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-D7/H4 | 101--116-D4-D8 | Enable adder sum to D-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| INCSC+ | 126-L3 | A | TL4 | L | (ACYLF+)(TL4FF+) | 126-H4 | 121-A5 | Increment shift counter |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |
| A1A24-D | 121-B5 | A | TL4 | S | (INCSC+)(SCZR0-)(MCSET+) | 121-A5 | 121-X6 | Step shift counter to ZERO |
| RRCXX- | 126-L6 | A | TL4 | L | (ACYEF+)(0PGWR+)(WRINH-) | 126-F6 | 150-D6 | Block STRB1+ to enable memory write cycle |
| DG0NE+ | 126-D9 | A | TL4 | L | (D01FF-) through (D16FF-) | 126-A9 | 125-D11 | D-register 1-16 equals ZERO |
| SETAZ+ | 125-J10 | A | TL4 | L | (0PGNS+)(ACYLF+)(DG0NE+) | 125-D11 | 125-J10 | Set AZZZZ+ F-F |
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| EDMTS+ | 128-H11 | A | TL1 | S | (RRCXX-)(0PGSM-)(TL1FF+) | 128-F11 | 101--116-G8 | Enable D-register to M-register |
| E1K17- | 127-L4 | A | TL3 | L | (ACYEF+)(1RS0P+)(AZZZZ+) | 127-E5 | 116-D7-D9 | Force carry to adder |
| EPSLL+ | 128-H4 | A | TL3 | L | (E01NS+)(1RS0P+) | 128-F4 | 101--116-A10 | Enable P-register to adder |
| CLDTR- | 125-J5 | A | TL3 | R | (ANA0P-)(TL3FF+)(MCRST+) | 125-A6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | A | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLPTR- | 129-H10 | A | TL4 | R | (0PGJS+)(E01NS+)(TL4FF+) | 129-D9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | A | TL4 | S | (0PGJS+)(E01NS+)(TL4FF+) | 129-D9 | 101--116-G11 | Enable D-register into P-register |
| CLYTR- | 129-J3 | A | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-D3 | 101--116-L11 | Clear Y-register |
| EDYTS+ | 129-L1 | A | TL4 | S | (MCSET+)(TL4FF+)(SCZR0-)(E01NS-)(0PGJS-) | 129-H1 H2 | 101--116-G10 | Enable D-register into Y-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

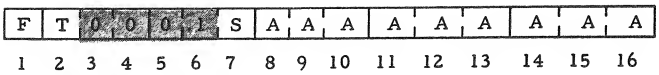

ENTER FROM PREVIOUS OPERATION
(FETCH CYCLE ENTRY ENABLE)
FCYEF+
T1
R 0 → (M), 1's → (D)
L (P) + 1 → ADDER, 0 → (F)
S ADDER → (D)
CLMTR-, CLDTR-
EPSLL+, E1K17-, CLFTL- (SEE NOTE)
ESDTS+
T2
STROBE [EA] → (M)
R 0 → (P)
S (D) → (P)
MMnnE-
CLPTR-
EDPTS+
L (M)3-6 → (F)3-6
EMFTL+
T2 & T3 (TLATE)
SECTOR ZERO? (M07 = 0)
YES
NO
INDEXING? (M02 = 1)
LDX?
(M)8-16 + (X)1-16 → (D) VIA ADDER
(Y)1-7, (M)8-16 + (X)1-16 → (D) VIA ADDER
L (X) → ADDER EXSTL+
L (M)1-7 → ADDER EMSHL+
L (M)8-16 → ADDER EMSLL+
L (M)-1-7 → ADDER ENSHL+
L (Y)1-7 → ADDER EYSHL+
0 → (D)1-7
(M)8-16 → (D)8-16 VIA ADDER
(Y)1-7, (M)8-16 → (D) VIA ADDER
T3
R 1's → (D) CLDTR-
S ADDER → (D) ESDTS+
(A) (TO SHEET 2)
JMP 1 CYCLE OP CODE 01 (SHEET 1)


(A) (FROM SHEET 1)
(FROM MEMORY REFERENCE F-CYCLE OR I-CYCLE. Page 2-4 or 2-6)
YES
INDIRECT ADDRESS? (M01 = 1)
NO ([EA] IS IN Y)
R 0 → (Y), 0 → (P) CLYTR-, CLPTR-
S (D) → (Y) EDYTS+
S (D) → (P) EDPTS+
L DEVELOP START MEMORY LEVEL MEMC1+
S SET F-CYCLE ENTRY F-F FCYEF+
STANDARD I-CYCLE
NEXT INSTRUCTION FETCH CYCLE
JMP 1 CYCLE OP CODE 01 (SHEET 2)

Instruction: Unconditional Jump (JMP)

OP Code: 01 Type: MR, 1 cycle

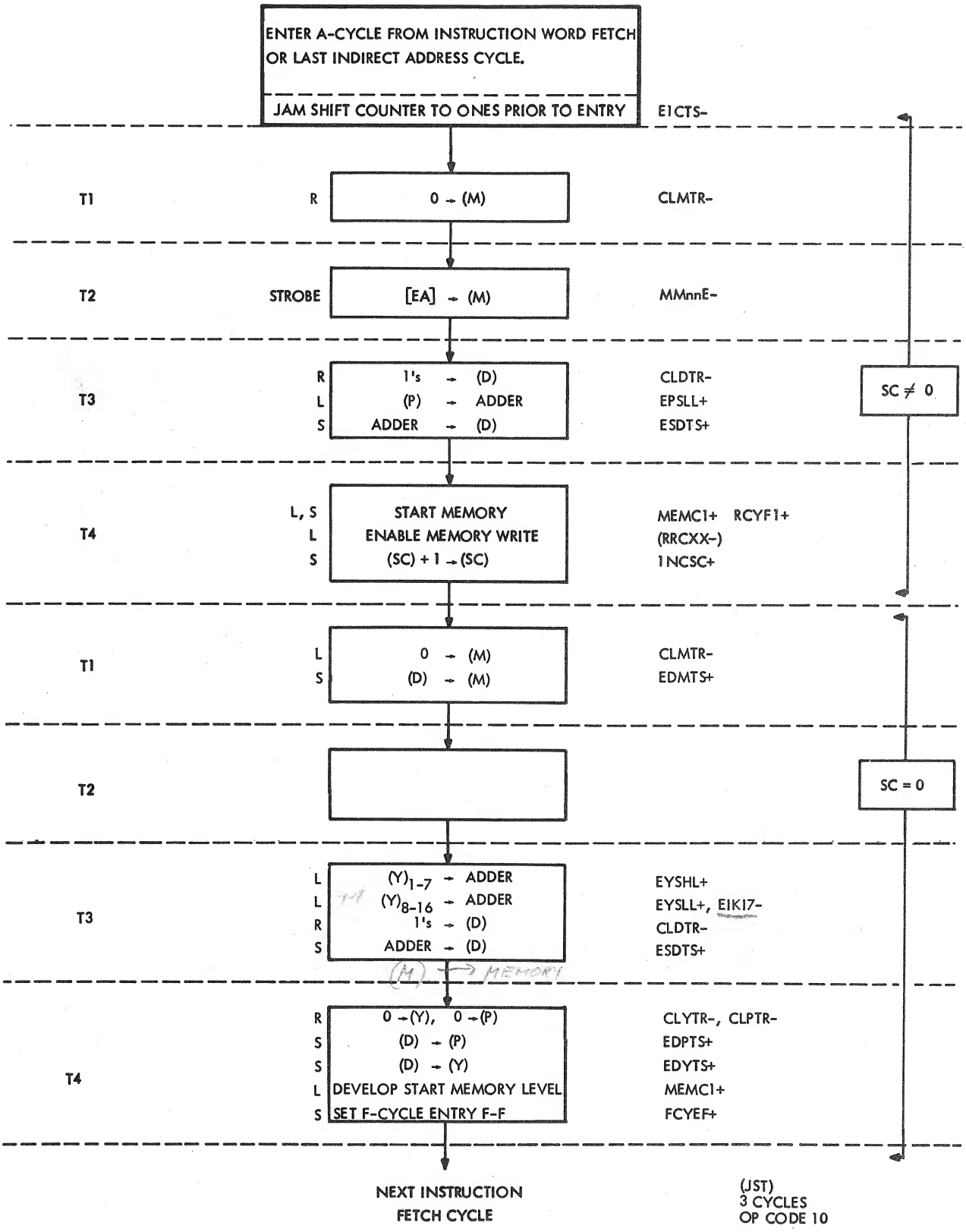
Description: EA → (P)




Execution Time (μsec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-J5 | 116-D7-D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) (MCRST+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) (MCRST+) | 125-A5 | 120-B1<br>121-A5<br>125-L10 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) (MCSET+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (EDPTL+)(MCRST+)* | 129-H10 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (EDPTL+)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register into P-register |
| EMFTL+ | 128-K3 | F | TLATE | L | [(M07FF+)∨(M07FF-)]∧ [(FCYLF+)(TLATE+)(GEN0P-) (MEMAC-)] | 127-G4 | 120-A2 | Enable M(3-6) into F(3-6) |
| EMFTL- | 128-K3 | F | TLATE | L | See EMFTL+ (M07FF+) | 127-G4 | 127-L9 | Generate EMSHL+ and ENSHL+ |
| EXSTL+ | 128-K5 | F | TLATE | L | (F01CY+)(MEMAC-)(M03FF-) (GEN0P-)(MEMAC-)(M03FF+) | 128-H6 | 101--116-A5 | Enable X-register to adder |
| EMSHL+ | 127-L8 | F | TLATE | L | (FCYS0-) | 127-G4 | 101--107-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (FCYS0-) | 127-G4 | 101--107-A9 | Enable M-(1-7) to adder |
| EMSLL+ | 127-L9 | F | TLATE | L | (FCYS0-) | 127-G4 | 108--116-A8 | Enable M(8-16) to adder |
| EYSHL+ | 128-K3 | F | TLATE | L | (FCYLF+)(TLATE+)(GEN0P-) (MEMAC-)(M07FF+) | 128-J3 | 101--107-A1 | Enable Y(1-7) to adder |
| CLDTR- | 125-J5 | F | TL3 | R | (ACYLF-)(TL3FF+)(MCRST+) | 125-D5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (10GRP-)(TL3FF+)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-A1/D3 | 101--116-L11 | Clear Y-register |
| CLPTR- | 129-J10 | F | TL4 | R | (0PGJS+)(E01NS+)(TL4FF+) (MCRST+) | 129-D9/H10 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL4 | S | (0PGJS+)(E01NS+)(TL4FF+) (MCSET+) | 129-D9/H9 | 101--116-H11 | Enable D-register into P-register |
| EDYTS+ | 129-L1 | F | TL4 | S | (ACYNX-)(MCSET+)(TL4FF+) (BREQ-)(0PGJS-) | 129-H1 | 101--116-G10 | Enable D-register into Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

*See gate A1A44-F at 129-D7 for EDPTL+

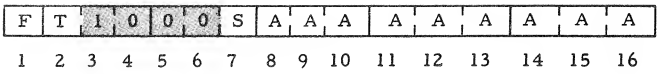



NOTE: FLAG AND TAG BITS ARE PRESERVED BY H02DJ+, 136-F6 FOR EXTENDED ADDRESSING

Instruction: Jump and Store Location (JST)

OP Code: 10 Type: MR, 3 cycles

Description: $(P)_{3-16} \rightarrow [EA]_{3-16}$
$[EA] + 1 \rightarrow (P)$



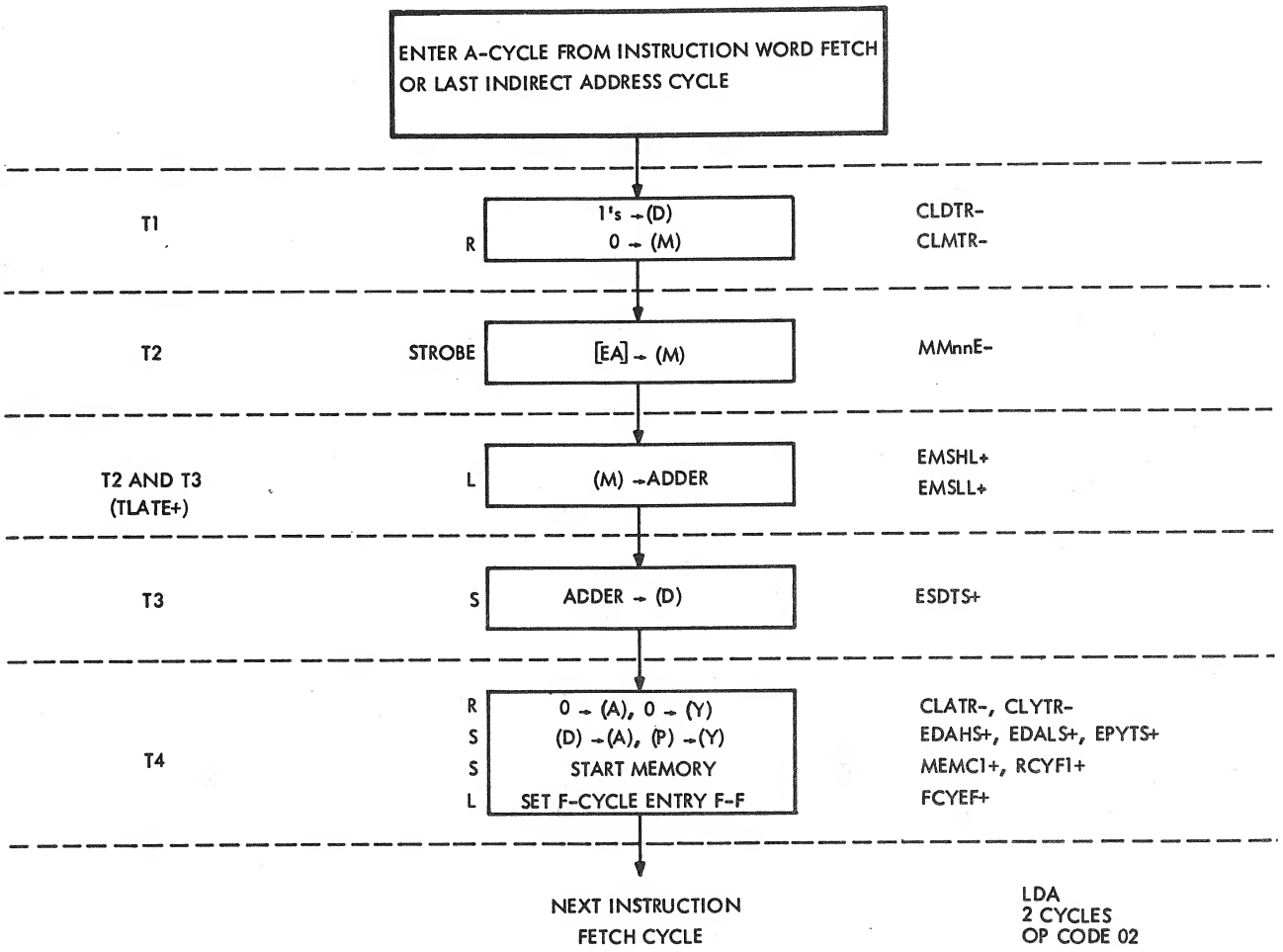

Execution Time ($\mu$sec): 2.88

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| ACYEF+ | 119-F4 | F | TL4 | L | (TL4FF+)(E01NS-)(F01CY+) [(M01FF-) at 119-B5] | 119-D5 | 119-H3 | Set A-cycle at next TL1 |
| E1CTS- | 125-J8 | F | TL4 | S | (FCYLF+)(TL4FF+)(0PG3C+) | 125-A7 | 121-A8 | ONEs to shift counter |
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLDTR- | 125-J5 | A | TL3 | R | (ANA0P-)(TL3FF+)(MCRST+) | 125-A6 | 101--116-E7 | Clear D-register to ONEs |
| EPSLL+ | 128-H4 | A | TL3 | L | (ACYEF+)(JST0P+)(SCZR0-) | 128-F5 | 101--116-A10 | Enable P-register to adder |
| ESDTS+ | 125-L4 | A | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (MEMC1-) | 126-G11 | 150-C1 | Enable set RCYF1+ |
| 1NCSC+ | 126-L3 | A | TL4 | L | (ACYLF+)(TL4FF+) | 126-H4 | 121-A5 | Enable step shift counter |
| A1A24-D | 121-B5 | A | TL4 | S | (1NCSC+)(SCZR0-)(MCSET+) | 121-A5 | 121-A5 | Shift counter to ZEROs |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | | Start memory cycle |
| RRCXX- | 126-L6 | A | TL4 | L | (ACYEF+)(0PGWR+)(WR1NH-) | 126-F6 | 150-D6 | Block STRB1+ and enable MWC* |
| | (See 1NCSC- for WR1NH- and JST0P- for 0PGWR+ at 126-D7 and 126-D6, respectively) | | | | | | | |
| CLMTR- | 128-K8 | A | TL1 | R | (See CLMTR- above) | | | |
| EDMTS+ | 128-H11 | A | TL1 | S | (RRCXX-)(0PGSM-)(MAST0-)(TL1FF+)(MCSET+) | 128-F11 | 101--116-G8 | Enable D-register into M-register |
| EYSHL+ | 128-K3 | A | TL3 | L | (ACYEF+)(JST0P+)(SCZR0+) | 128-H2 | 101--107-A11 | Y(1-7) to adder |
| EYSLL+ | 128-K2 | A | TL3 | L | (ACYEF+)(JST0P+)(SCZR0+) | 128-H2 | 108--116-A11 | Y(8-16) to adder |
| E1K17- | 127-L4 | A | TL3 | L | (EYSLL-) | 127-J5 | 116-D7 | Force carry to adder |
| CLDTR- | 125-J5 | A | TL3 | R | (ANA0P-)(TL3FF+)(MCRST+) | 125-A6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | A | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLYTR- | 129-J3 | A | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-A1/D3 | 101--116-L11 | Clear Y-register |
| CLPTR- | 129-J10 | A | TL4 | R | (0PGJS+)(E01NS+)(TL4FF+)(MCRST+) | 129-D7/H10 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | A | TL4 | S | (0PGJS+)(E01NS+)(TL4FF+)(MCSET+) | 129-D7/H9 | 101--116-H11 | Enable D-register into P-register |
| EDYTS+ | 129-L1 | A | TL4 | S | (ACYNX-)(MCSET+)(TL4FF+)(BRREQ-)(0PGJS-) | 129-H1 | 101--116-G10 | Enable D-register into Y-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/A11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

SC ≠ 0

SC = 0

*MWC = Memory write cycle

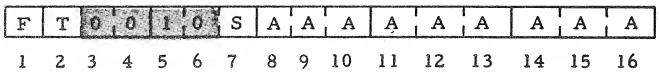

ENTER A-CYCLE FROM INSTRUCTION WORD FETCH
OR LAST INDIRECT ADDRESS CYCLE
T1
R
1's → (D)
0 → (M)
CLDTR-
CLMTR-
T2
STROBE
[EA] → (M)
MMnnE-
T2 AND T3
(TLATE+)
L
(M) → ADDER
EMSHL+
EMSLL+
T3
S
ADDER → (D)
ESDTS+
T4
R
S
S
L
0 → (A), 0 → (Y)
(D) → (A), (P) → (Y)
START MEMORY
SET F-CYCLE ENTRY F-F
CLATR-, CLYTR-
EDAHS+, EDALS+, EPYTS+
MEMC1+, RCYF1+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
LDA
2 CYCLES
OP CODE 02

3396

Instruction: Load A (LDA)

OP Code: 02 Type: MR, 2 cycle

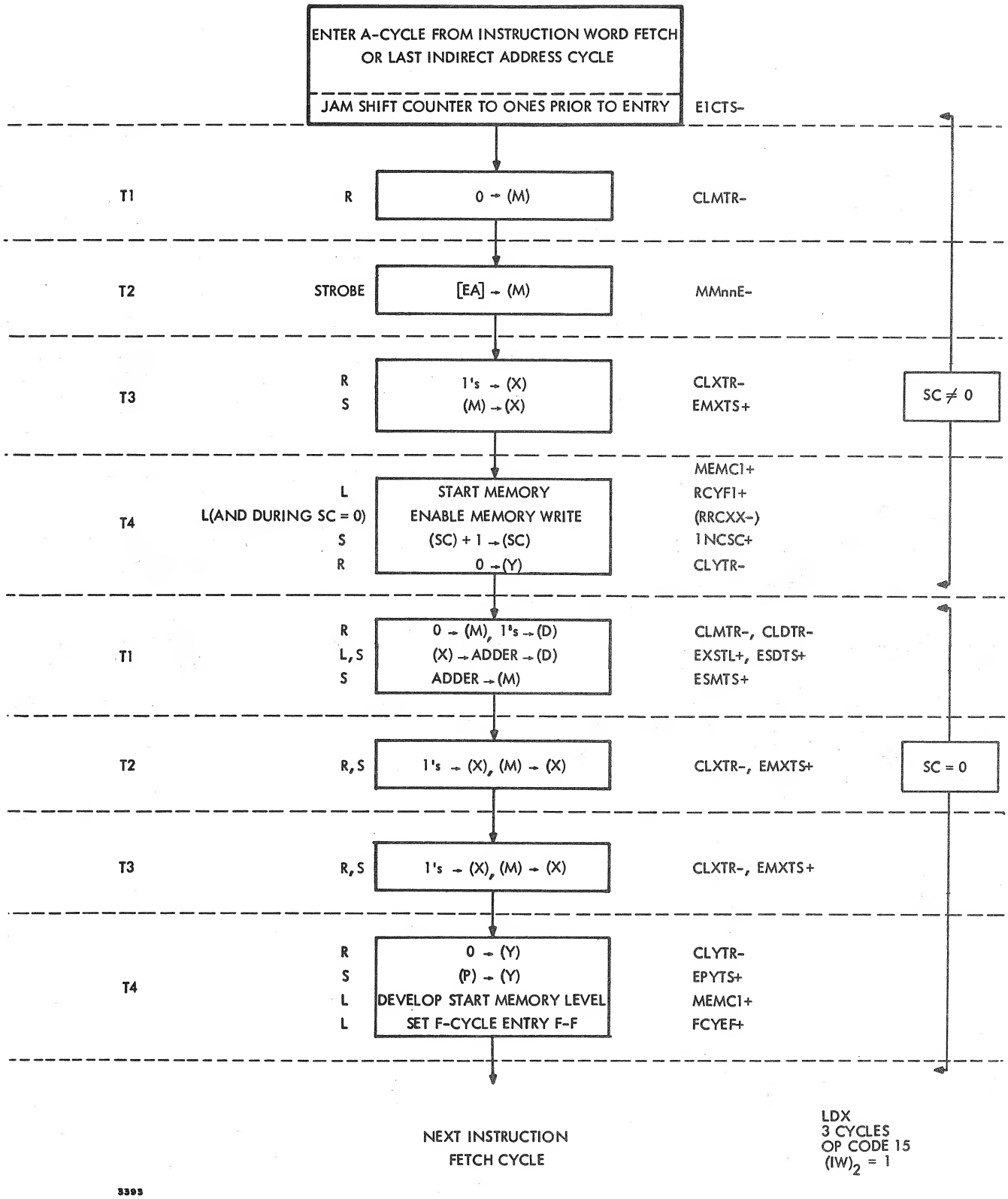
Description: (EA) → (A)




Execution Time (μsec): 1.92

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| ACYEF+ | 119-F4 | F | TL4 | L | (TL4FF+)(E01NS-)(F01CY+) [(M01FF-) @ 119-B5] | 119-D5 | 119-H3 | Set A-cycle at next TL1 |
| CLDTR- | 125-J6 | A | TL1 | R | (ACYEF+)(TL1FF+)(JST0P-) (1RS0P-)(1MA0P-) | 125-D4 | 101--116-E7 | Clear D-register to ONEs |
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM+)(TL1FF+) | 128-K8 | 101--116-H9 | Reset M-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| EMSHL+ | 127-L8 | A | TLATE | | [(ACYLF+)(TLATE+) | 127-G8 | 101--108-A8 | Enable M(1-7) to adder |
| EMSLL+ | 127-L10 | A | TLATE | | (SUB0P-)(0PGAA+)] | 127-G8 | 109--116-A8 | Enable M(8-16) to adder |
| ESDTS+ | 125-L4 | A | TL3 | S | (TL3FF+)(10GRP-) | 125-D6 | 101--116-D4/D6/D8 | Enable adder sum to D-register |
| CLATR- | 122-J7 | A | TL4 | R | (CLATL+)(MCRST+)* | 122-H7 | 101--116-H5 | Reset A-register |
| CLYTR- | 129-J3 | A | TL4 | R | (TL4FF+)(ACYNX-)** | 129-D3 | 101--116-L11 | Reset Y-register |
| EDAHS+ | 122-L1 | A | TL4 | S | [(ACYLF+)(TL4FF+) | 122-D3 | 101--108-G7 | Enable D(1-8) into A(1-8) |
| EDALS+ | 122-L2 | A | TL4 | S | (0PGAA+)(1MA0P-)] | 122-D3 | 109--116-G7 | Enable D(9-16) into A(9-16) |
| EPYTS+ | 129-L5 | A | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+) (0PGJS-) | 129-D4 | 101--116-J11 | Enable P-register into Y-register |
| FCYEF+ | 119-F10 | A | TL4 | L | [Set: (TL4FF+)(E01NS+) (DMCRQ-)(P1SEX-)] | 119-D10 | 119-H10 | Enable set FCYLF+ at next TL1FF+ |
| MEMC1+ | 126-J11 | A | TL4 | L | (MEMC1-) | 126-G11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+) (RCYF1-) | 150-C2 | XXXX | Start memory cycle |

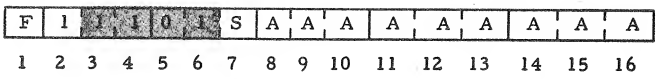
*(CLATL+) @ 122-F7 = (ACYLF+)(TL4FF+)(0PGAA+)(1MA0P-) @ 122-D3
**(ACYNX-) @ 129-B1 = ((ACYLF+)(LSX0P-) (CAS0P-)(SCZR0-))- @ 129-A1


ENTER A-CYCLE FROM INSTRUCTION WORD FETCH OR LAST INDIRECT ADDRESS CYCLE
JAM SHIFT COUNTER TO ONES PRIOR TO ENTRY
E1CTS-
T1
R
0 → (M)
CLMTR-
T2
STROBE
[EA] → (M)
MMnnE-
T3
R
S
1's → (X)
(M) → (X)
CLXTR-
EMXTS+
SC ≠ 0
T4
L
L(AND DURING SC = 0)
S
R
START MEMORY
ENABLE MEMORY WRITE
(SC) + 1 → (SC)
0 → (Y)
MEMC1+
RCYF1+
(RRCXX-)
1NCSC+
CLYTR-
T1
R
L, S
S
0 → (M), 1's → (D)
(X) → ADDER → (D)
ADDER → (M)
CLMTR-, CLDTR-
EXSTL+, ESDTS+
ESMTS+
T2
R, S
1's → (X), (M) → (X)
CLXTR-, EMXTS+
SC = 0
T3
R, S
1's → (X), (M) → (X)
CLXTR-, EMXTS+
T4
R
S
L
L
0 → (Y)
(P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY F-F
CLYTR-
EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION FETCH CYCLE
LDX
3 CYCLES
OP CODE 15
(IW)2 = 1
3393

Instruction: Load Index Register (LDX)

OP Code: 15 Type: MR, 3 cycle

Description: (EA) → X
X → (0) Note: Bit 2 of IW must be set

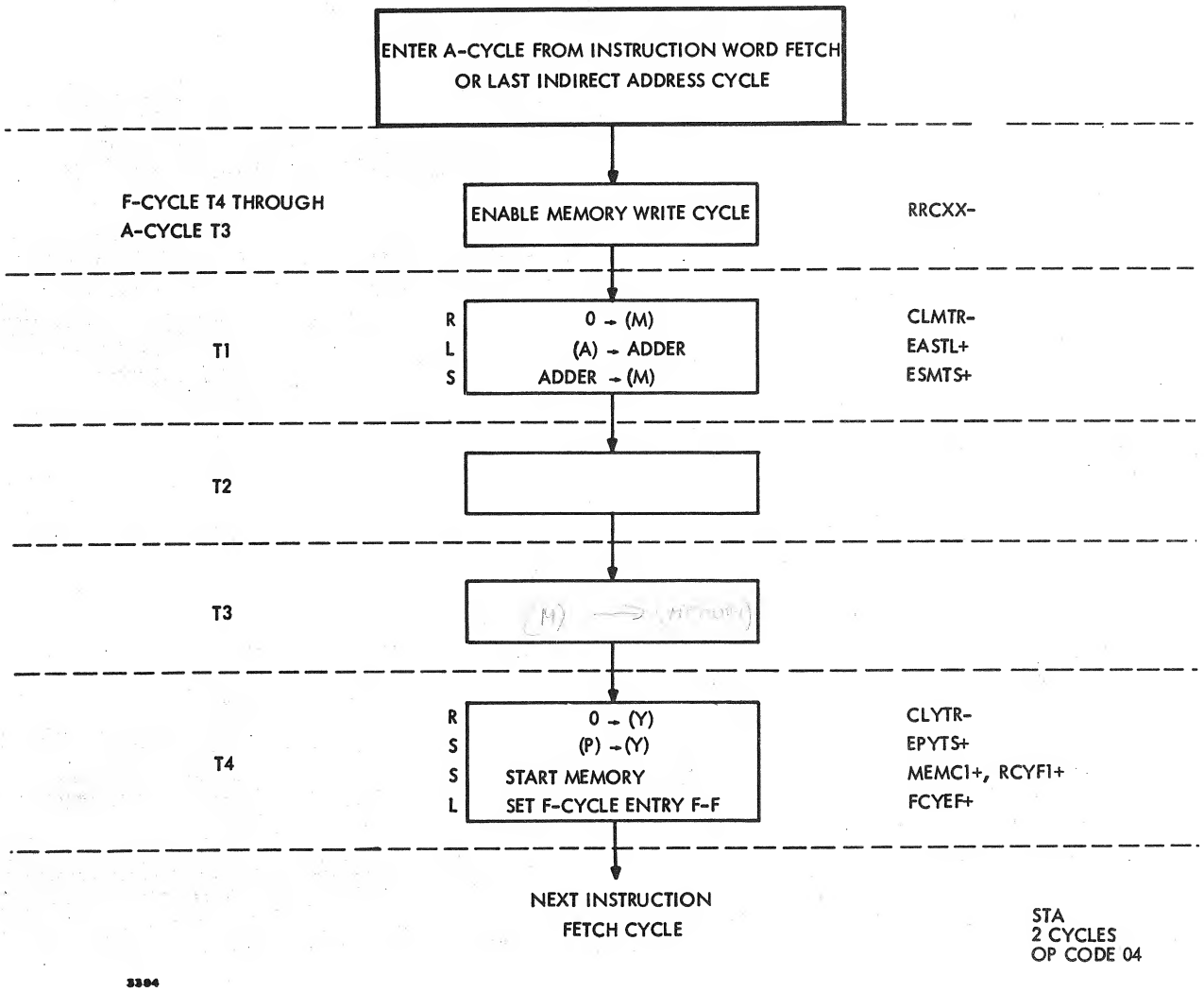



Execution Time (μsec): 2.88

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| ACYEF+ | 119-F4 | F | TL4 | L | (TL4FF+)(E01NS-)(F01CY+) [(M01FF-) @ 119-B5] | 119-D4 | 119-H3 | Set A-cycle at next TL1 |
| E1CTS- | 125-J8 | F | TL4 | S | (FCYLF+)(TL4FF+)(0PG3C+)* | 125-A7 | 121-A8 | Jam shift counter to ONEs |
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Reset M-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLXTR- | 128-F8 | A | TL3 | R | (ACYLF+)(TL3FF+)(LSX0P+) | 128-B9 | 101--116-L8 | Clear X-register to ONEs |
| EMXTS+ | 128-H7 | A | TL3 | S | (ACYLF+)(TL3FF+)(LSX0P+) | 128-B9 | 101--116-J9 | Enable M-register into X-register |
| 1NCSC- | 126-J3 | A | TL4 | L | (ACYLF+)(TL4FF+) | 126-H4 | 126-D7 | Reset WR1NH+ F-F |
| 1NCSC+ | 126-L3 | A | TL4 | L | (1NCSC-) | 126-J3 | 121-A5 | Enable step shift counter from all ONEs to ZEROs |
| MEMC1+ | 126-J11 | A | TL4 | L | (MEMC1-) | 126-G11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | XXXX | Start memory cycle |
| RRCXX- | 126-L6 | A | ** | L | (ACYEF+)(0PGWR+)(WR1NH-) | 126-F6 | 150-D6 | Block STRB1+ to enable memory write cycle |
| CLYTR- | 129-J3 | A | TL4 | R | (ACYNX-)(TL4FF+) | 129-D3 | 101--116-L11 | Reset Y-register (address location ZERO in memory) |
| EXSTL+ | 128-K5 | A | TL1 | L | (ACYEF+)(TLATE-)(LSX0P+) | 128-H5 | 101--116-A5 | Enable X-register to adder |
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Reset M-register |
| CLDTR- | 125-J5 | A | TL1 | R | (ACYEF+)(TL1FF+)(JST0P-) (1RS0P-)(1MA0P-) | 125-D4 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | A | TL1 | S | (ACYEF+)(TL1FF+)(JST0P-) (1RS0P-)(1MA0P-) | 125-D4 | 101--116-D4/D6/D8 | Enable adder sum into D-register |
| ESMTS+ | 128-H10 | A | TL1 | S | (RRCXX-)(MAST0-)(0PGSM+) (TL1FF+)(MCSET+) | 128-F10 | 101--116-G9 | Enable adder RnnPA+ output into M-register |
| CLXTR- | 128-F8 | A | TL3 | R | (ACYLF+)(TL3FF+)(LSX0P+) | 128-B9 | 101--116-L8 | Clear X-register to ONEs |
| EMXTS+ | 128-H7 | A | TL3 | S | (ACYLF+)(TL3FF+)(LSX0P+) | 128-B9 | 101--116-J9 | Enable M-register into X-register |
| CLYTR- | 129-J3 | A | TL4 | R | (TL4FF+)(ACYNX-)*** | 129-D3 | 101--116-L11 | Reset Y-register |
| FCYEF+ | 119-F10 | A | TL4 | L | (Set: (TL4FF+)(E01NS+) | 119-D10 | 119-H9 | Enable set FCYLF+ at next TL1FF+ |
| EPYTS+ | 129-L5 | A | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+) (0PGJS-) | 129-D4 | 101--116-J11 | Enable P-register into Y-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (MEMC1-) | 126-G11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | XXXX | Start memory cycle |

*(0PG3C+) @ 120-F9 = (DP0LX-), etc.
(DP0LX-) = (LSX0P+)(M02FF+) @ 124-H11
**See WR1NH Set and reset timing

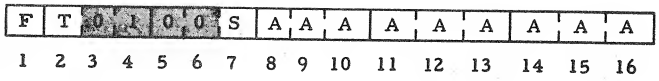
***(ACYNX-) = ((ACYLF+)(LSX0P-)(CAS0P-)(SCZR0-)) - @ 129-A1


ENTER A-CYCLE FROM INSTRUCTION WORD FETCH
OR LAST INDIRECT ADDRESS CYCLE
F-CYCLE T4 THROUGH
A-CYCLE T3
ENABLE MEMORY WRITE CYCLE
RRCXX-
T1
R
L
S
0 → (M)
(A) → ADDER
ADDER → (M)
CLMTR-
EASTL+
ESMTS+
T2
T3
T4
R
S
S
L
0 → (Y)
(P) → (Y)
START MEMORY
SET F-CYCLE ENTRY F-F
CLYTR-
EPYTS+
MEMC1+, RCYF1+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
STA
2 CYCLES
OP CODE 04
3304

Instruction: Store A (STA)

OP Code: 04 Type: MR, 2-cycle

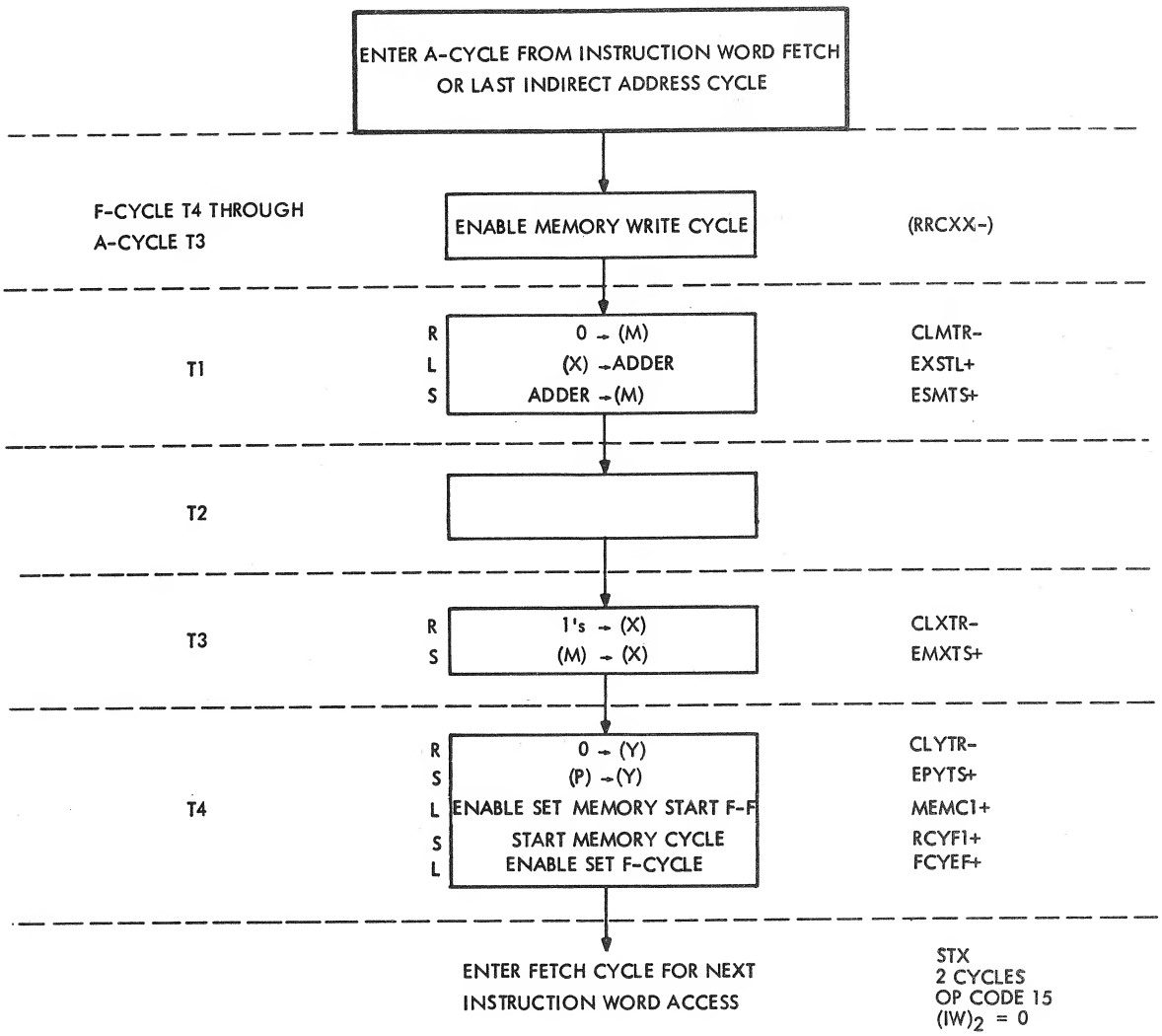
Description: (A) → (EA)




Execution Time ($\mu$sec): 1.92

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| ACYEF+ | 119-F4 | F | TL4 | L | (TL4FF+)(E01NS-)(F01CY+) [(M01FF-) @ 119-B5] | 119-D5 | 119-H3 | Set A-cycle at next TL1 |
| RRCXX- | 126-L6 | F<br>A | TL4 thru<br>TL3 | L<br>L | (ACYEF+)(0PGWR+)(WR1NH-) | 126-F6 | 150-D6 | Block STRB1+ to enable memory write cycle |
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Reset M-register |
| EASTL+ | 127-L1 | A | TL1 | L | (ACYEF+)(TLATE-)(CAS0P-) (LSX0P-)(10GRP-) | 127-J1 | 101--116-A5 | Enable A-register to adder |
| ESMTS+ | 128-H10 | A<br>A<br>A | TL1<br>TL2<br>TL3 | S<br>L<br>L | (RRCXX-)(MAST0-)(0PGSM+) (TL1FF+)(MCSET+) | 128-F10 | 101--116-G9 | Enable adder RnnPA+ output into M-register |
| CLYTR- | 129-J3 | A | TL4 | R | (TL4FF+)(ACYNX-)* | 129-D3 | 101--116-L11 | Reset Y-register |
| EPYTS+ | 129-L5 | A | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+) (0PGJS-) | 129-D4 | 101--116-J11 | Enable P-register into Y-register |
| FCYEF+ | 119-F10 | A | TL4 | L | (Set: (TL4FF+)(E01NS+)) | 119-D10 | 119-H10 | Enable set FCYLF+ at next TL1FF+ |
| MEMC1+ | 126-J11 | A | TL4 | L | (MEMC1-) | 126-G11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+) (RCYF1-) | 150-C2 | XXXX | Start memory cycle |

*(ACYNX-) = ((ACYLF+)(LSX0P-)(CAS0P-)(SCZR0-))- @ 129-A1

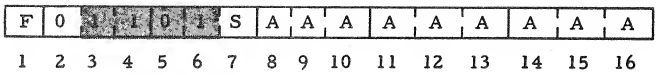

ENTER A-CYCLE FROM INSTRUCTION WORD FETCH
OR LAST INDIRECT ADDRESS CYCLE
F-CYCLE T4 THROUGH
A-CYCLE T3
ENABLE MEMORY WRITE CYCLE
(RRCXX-)
T1
R
L
S
0 → (M)
(X) → ADDER
ADDER → (M)
CLMTR-
EXSTL+
ESMTS+
T2
T3
R
S
1's → (X)
(M) → (X)
CLXTR-
EMXTS+
T4
R
S
L
S
L
0 → (Y)
(P) → (Y)
ENABLE SET MEMORY START F-F
START MEMORY CYCLE
ENABLE SET F-CYCLE
CLYTR-
EPYTS+
MEMC1+
RCYF1+
FCYEF+
ENTER FETCH CYCLE FOR NEXT
INSTRUCTION WORD ACCESS
STX
2 CYCLES
OP CODE 15
(IW)2 = 0

3399

Instruction: Store Index Register (STX)

OP Code: 15 Type: MR, 2-cycle

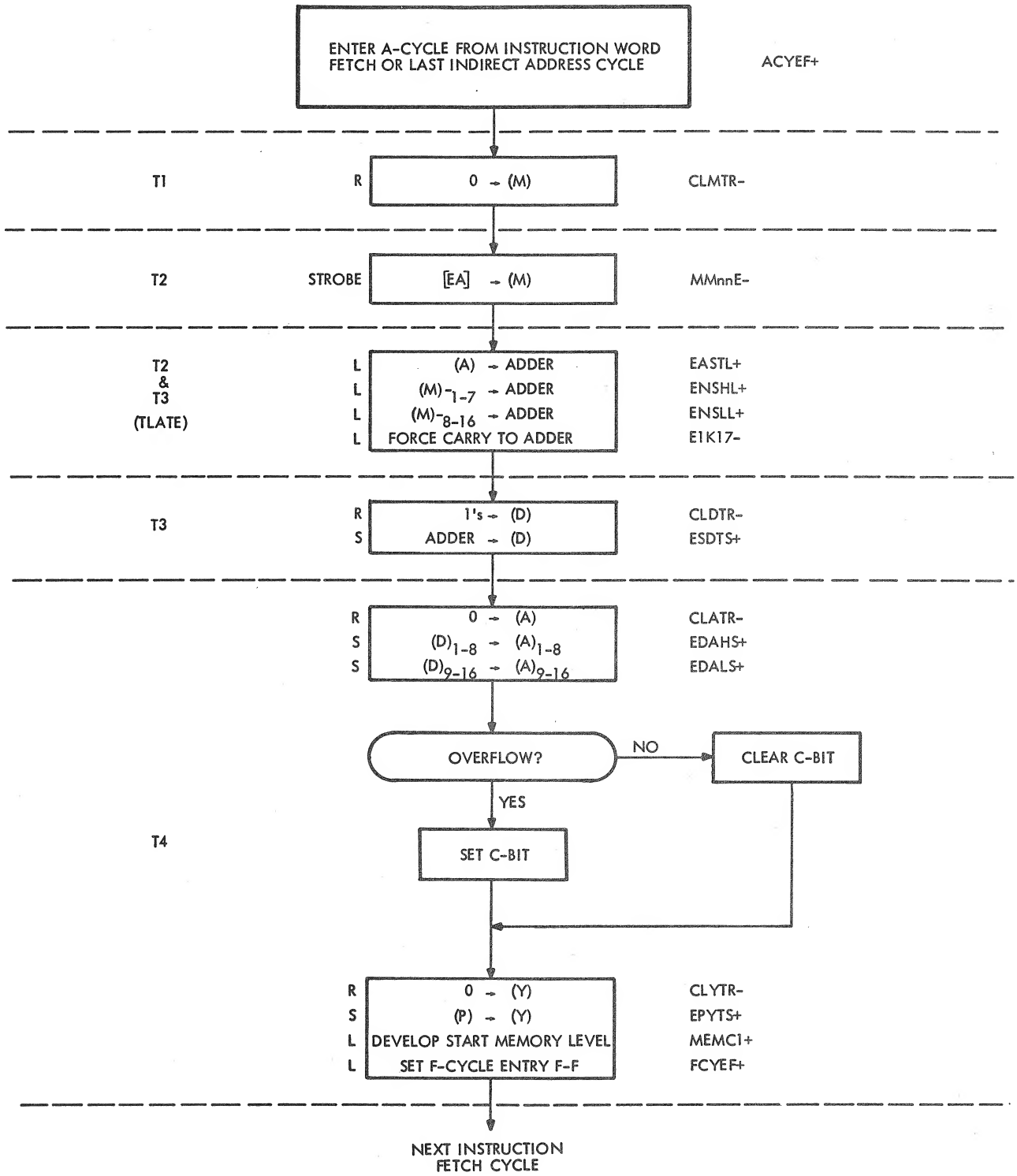
Description: (X) → (EA) Note: Bit 2 of 1W must be reset




Execution Time (μsec): 1.92

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| ACYEF+ | 119-F4 | F | TL4 | L | (TL4FF+)(E01NS-)(F01CY) [(M01FF-) @ 119-B5] | 119-D5 | 119-H3 | Set A-cycle at next TL1 |
| RRCXX- | 126-L6 | F<br>A | TL4<br>TL3 | L-<br>L | (ACYEF+)(0PGWR+)(WR1NH-) | 126-F6 | 150-D6 | Block STRB1+ to enable memory write cycle |
| EXSTL+ | 128-K5 | A | TL1 | L | (ACYEF+)(TLATE-)(LSX0P+) | 128-H5 | 101--116-A5 | Enable X-register to adder |
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 121-K8 | 101--116-H9 | Reset M-register |
| ESMTS+ | 128-H10 | A | TL1 | S | (RRCXX-)(MAST0-)(0PGSM+) (TL1FF+)(MCSET+) | 128-F10 | 101--116-G9 | Enable adder RnnPA+ output into M-register |
| CLXTR- | 128-F8 | A | TL3 | R | (ACYLF+)(TL3FF+)(LSX0P+) | 128-B8 | 101--116-L8 | Clear X-register to ONEs |
| EMXTS+ | 128-H7 | A | TL3 | S | (ACYLF+)(TL3FF+)(LSX0P+) | 128-B8 | 101--116-J9 | Enable M-register into X-register |
| CLYTR- | 129-J3 | A | TL4 | R | (TL4FF+)(ACYNX-)* | 129-D3 | 101--116-L11 | Reset Y-register |
| FCYEF+ | 119-F10 | A | TL4 | L | (Set: (TL4FF+)(E01NS+)) | 119-D10 | 119-H9 | Enable set FCYLF+ at next TL1FF+ |
| EPYTS+ | 129-L5 | A | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+) (0PGJS-) | 129-D4 | 101--116-J11 | Enable P-register into Y-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (MEMC1-) | 126-G11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | XXXX | Start memory cycle |

*(ACYNX-) = ((ACYLF+)(LSX0P-)(CAS0P-)(SCZR0-))- @ 129-A1



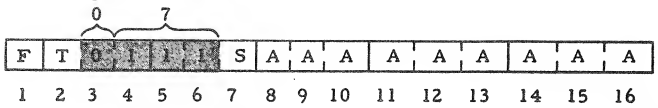

SUB
2 CYCLES
OP CODE 07

3498

Instruction: Subtract (SUB)

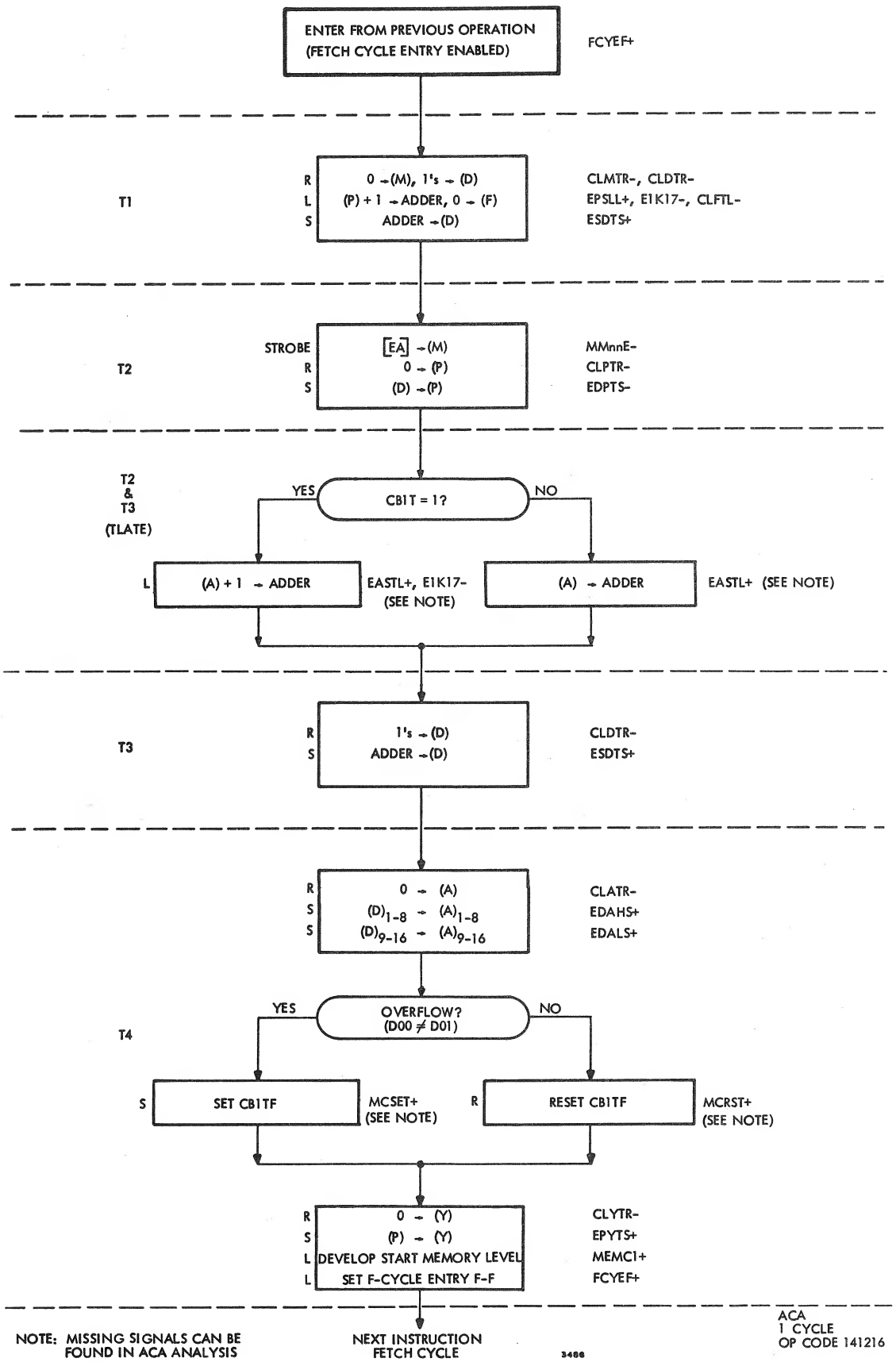
OP Code: 07 Type: MR, 2 cycles

Description: (A) − (EA) → (A)
OVF → (C)




Execution Time (μsec): 1.92

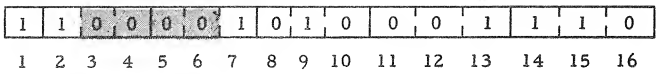
| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| CLMTR- | 128-K8 | A | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| EASTL+ | 127-L1 | A | TLATE | L | [(SUB0P-)]-(ACYLF+)(TLATE+) | 127-A1/C1 | 101--116-A5 | Enable A-register to adder |
| ENSHL+ | 127-L7 | A | TLATE | L | (ACYLF+)(0PGNS+)(1RS0P-) | 127-C11 | 101--107-A9 | Enable M-(1-7) to adder |
| ENSLL+ | 127-L5 | A | TLATE | L | (ACYLF+)(0PGNS+)(1RS0P-) | 127-C11 | 108--116-A9 | Enable M-(8-16) to adder |
| E1K17- | 127-L4 | A | TLATE | L | (See gate A1B55-E) | 127-J6 | 116-D7-D9 | Force carry to adder |
| CLDTR- | 125-J5 | A | TL3 | R | (ANA0P-)(TL3FF+)(MCRST+) | 125-A6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | A | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLATR- | 122-H7 | A | TL4 | R | (ACYLF+)(0PGAA+)(TL4FF+) | 122-D3 | 101--116-H5 | Clear A-register |
| EDAHS+ | 122-L1 | A | TL4 | S | (ACYLF+)(0PGAA+)(TL4FF+) | 122-D3 | 101--108-G7 | Enable D-register to A(1-8) |
| EDALS+ | 122-L2 | A | TL4 | S | (ACYLF+)(0PGAA+)(TL4FF+) | 122-D3 | 109--116-G7 | Enable D-register to A (9-16) |
| CB1TF+ | 124-L2 | A | TL4 | S | (SUB0P+)(TL4FF+)(D00 = D01)(MCSET+) | 124-A4 | 124-L2 | Set CB1TF |
| CB1TF- | 124-L1 | A | TL4 | R | (SUB0P+)(D1V0P-)(TL4FF+)(MCRST+) | 124-A1 | 124-L1 | Reset CB1TF |
| CLYTR- | 129-J3 | A | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-A1 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | A | TL4 | S | (P1SEX-)(E01NS+)(0PGJS-) | 129-D4/H4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11 H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |


ENTER FROM PREVIOUS OPERATION
(FETCH CYCLE ENTRY ENABLED)
FCYEF+
T1
R
L
S
0 → (M), 1's → (D)
(P) + 1 → ADDER, 0 → (F)
ADDER → (D)
CLMTR-, CLDTR-
EPSLL+, E1K17-, CLFTL-
ESDTS+
T2
STROBE
R
S
[EA] → (M)
0 → (P)
(D) → (P)
MMnnE-
CLPTR-
EDPTS-
T2
&
T3
(TLATE)
YES
CB1T = 1?
NO
L
(A) + 1 → ADDER
EASTL+, E1K17-
(SEE NOTE)
(A) → ADDER
EASTL+ (SEE NOTE)
T3
R
S
1's → (D)
ADDER → (D)
CLDTR-
ESDTS+
R
S
S
0 → (A)
(D)1-8 → (A)1-8
(D)9-16 → (A)9-16
CLATR-
EDAHS+
EDALS+
T4
YES
OVERFLOW?
(D00 ≠ D01)
NO
S
SET CB1TF
MCSET+
(SEE NOTE)
R
RESET CB1TF
MCRST+
(SEE NOTE)
R
S
L
L
0 → (Y)
(P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY F-F
CLYTR-
EPYTS+
MEMC1+
FCYEF+
NOTE: MISSING SIGNALS CAN BE FOUND IN ACA ANALYSIS
NEXT INSTRUCTION FETCH CYCLE
3466
ACA
1 CYCLE
OP CODE 141216

Instruction: Add C to A (ACA)

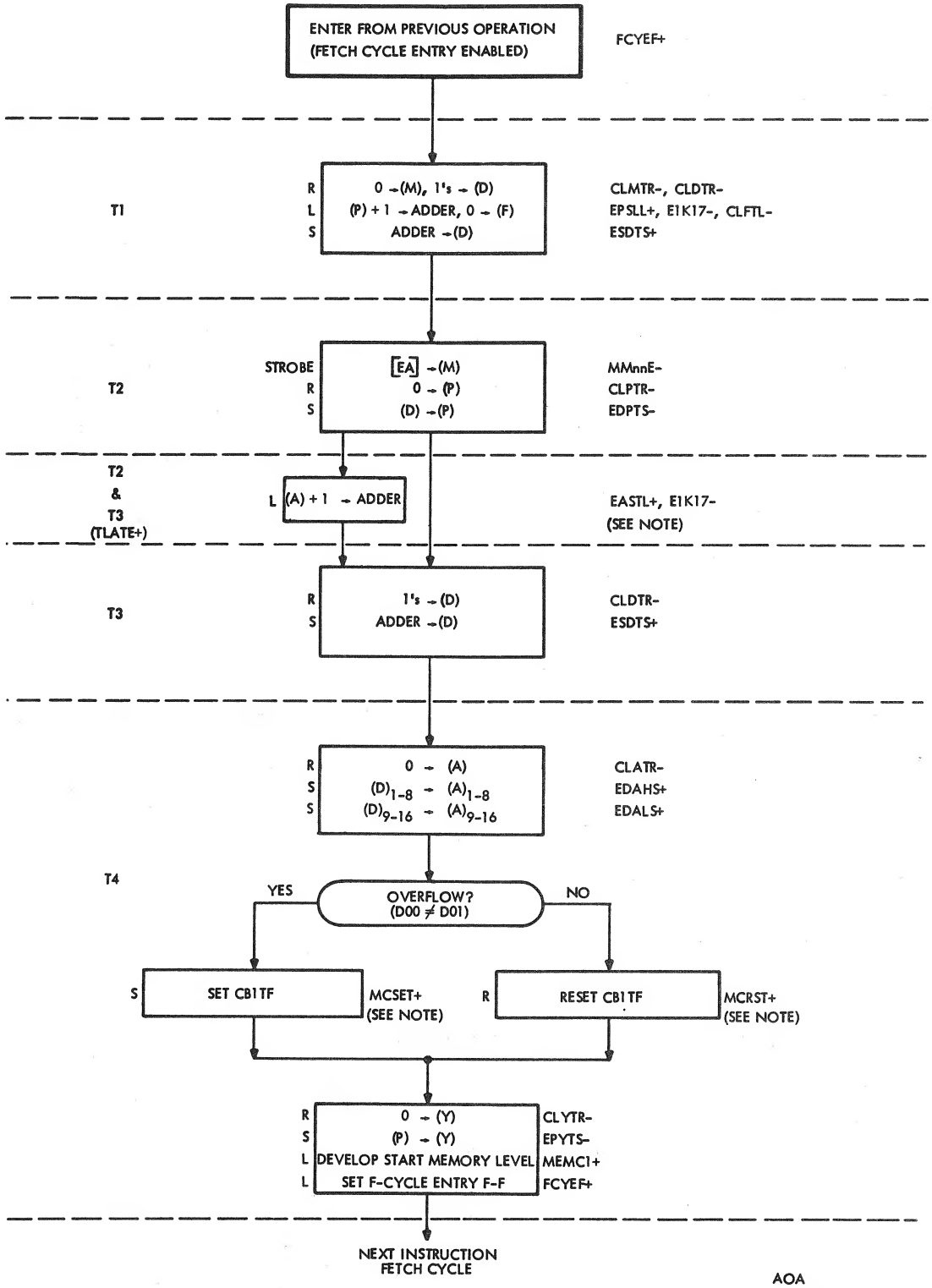
OP Code: 141216 Type: G, 1 cycle

Description: (A) + (C) → (A)
OVF → (C)




Execution Time (μsec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-J5 | 116-D7/D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCRST+) | 129-D7/H9 | 101--116-H10 | Clear P-register |
| E1K17- | 127-L4 | F | TLATE | L | (CB1TF+)(GEN0P+)(M01FF+)(M15FF+) | 127-C10/E4/J6 | 116-D7<br>117-B1 | Force carry to adder |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCSET+) | 129-H9 | 101--116-D4-D8 | Enable adder sum to D-register |
| EASTL+ | 127-L1 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--116-A5 | Enable A-register to adder |
| EMSHL+ | 127-L8 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--107-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--107-A9 | Enable M-(1-7) to adder |
| EMSLL+ | 127-L10 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 108--116-A8 | Enable M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 108--116-A9 | Enable M-(8-16) to adder |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLATR- | 122-H7 | F | TL4 | R | (M15FF+)(GEN0A+)(TL4FF+) | 122-A3 | 101--116-H5 | Clear A-register |
| EDAHS+ | 122-L1 | F | TL4 | S | (M15FF+)(GEN0A+)(TL4FF+) | 122-A3 | 101--108-G7 | Enable D-register to A(1-8) |
| EDALS+ | 122-L2 | F | TL4 | S | (M15FF+)(GEN0A+)(TL4FF+) | 122-A3 | 109--116-G7 | Enable D-register to A(9-16) |
| CB1TF- | 124-L1 | F | TL4 | R | (TL4FF+)(D1V0P-)(GEN0A+)(M09FF+)(M11FF-)(MCRST+) | 124-B1 | 124-L1 | Clear CB1TF at TL4 with reset clock |
| CB1TF+ | 124-L2 | F | TL4 | S | (D00 ≠ D01)(TL4FF+)(GEN0A+)(M09FF+)(M11FF-)(MCSET+) | 124-B2/C2 | 124-L2 | Set CB1TFF when D00 ≠ D01 |
| CLYTR- | 129-J3 | F | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-D2 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |




AOA
1 CYCLE
OP CODE 141206

NOTE: MISSING SIGNALS CAN BE FOUND IN AOA ANALYSIS

3497

Instruction: Add ONE to A (AOA)

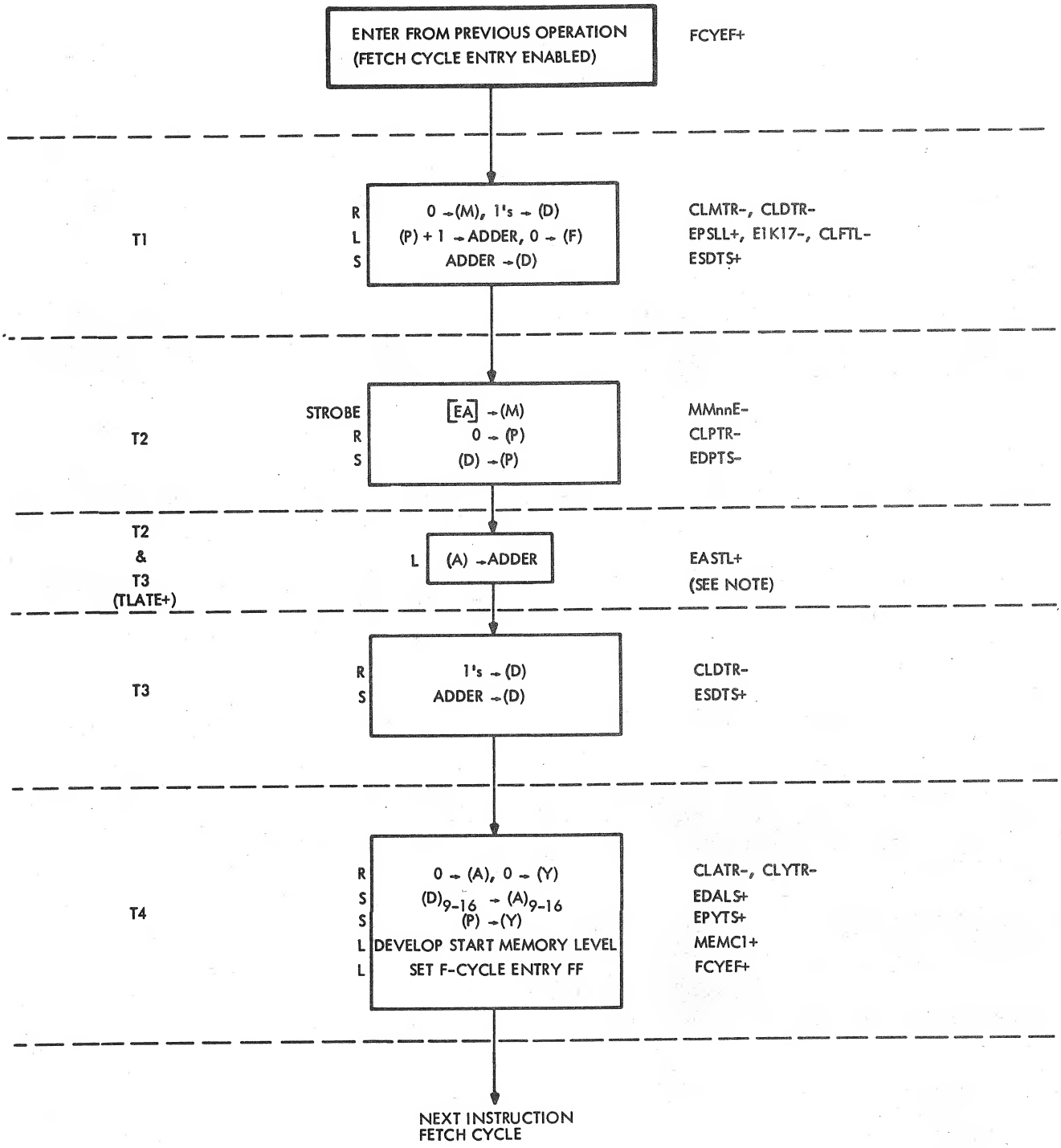
OP Code: 141206 Type: G, 1 cycle

Description: (A) + 1 → (A)
OVF → (C)

| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

Execution Time (μsec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-J5 | 116-D7/D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCRST+) | 129-D7/H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| EASTL+ | 127-L1 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--116-A5 | Enable A-register to adder |
| EMSHL+ | 127-L8 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--107-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--107-A9 | Enable M-(1-7) to adder |
| EMSLL+ | 127-L10 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 108--116-A8 | Enable M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 108--116-A9 | Enable M-(8-16) to adder |
| E1K17- | 127-L4 | F | TLATE | L | (M13FF-)(GEN0P+)(M01FF+)(M15FF+) | 127-C10/E4/J6 | 116-D7<br>117-B1 | Force carry to adder |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLATR- | 122-F7 | F | TL4 | R | (M15FF+)(GEN0A+)(TL4FF+) | 122-A3 | 101--116-H5 | Clear A-Register |
| EDAHS+ | 122-L1 | F | TL4 | S | (M15FF+)(GEN0A+)(TL4FF+) | 122-A3 | 101--108-G7 | Enable D-register to A(1-8) |
| EDALS+ | 122-L2 | F | TL4 | S | (M15FF+)(GEN0A+)(TL4FF+) | 122-A3 | 109--116-G7 | Enable D-register to A(9-16) |
| CB1TF- | 124-L1 | F | TL4 | R | (TL4FF+)(D1V0P-)(GEN0A+)-(M09FF+)(M11FF-)(MCRST+) | 124-A2 | 124-L1 | Clear C-bit with reset clock |
| CB1TF+ | 124-L2 | F | TL4 | S | (D00 ≠ D01)(GEN0A+)(M09FF+)(M11FF-)(TL4FF+)(MCSET+) | 124-B2/C2 | 132-C7 | Set C-bit when D00 ≠ D01 |
| CLYTR- | 129-J3 | F | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-D2 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |




CAL
1 CYCLE
OP CODE 141050

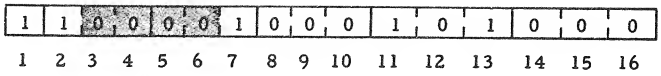
NOTE: MISSING SIGNALS CAN BE FOUND IN CAL ANALYSIS

3401

Instruction: Clear Left Half (CAL)

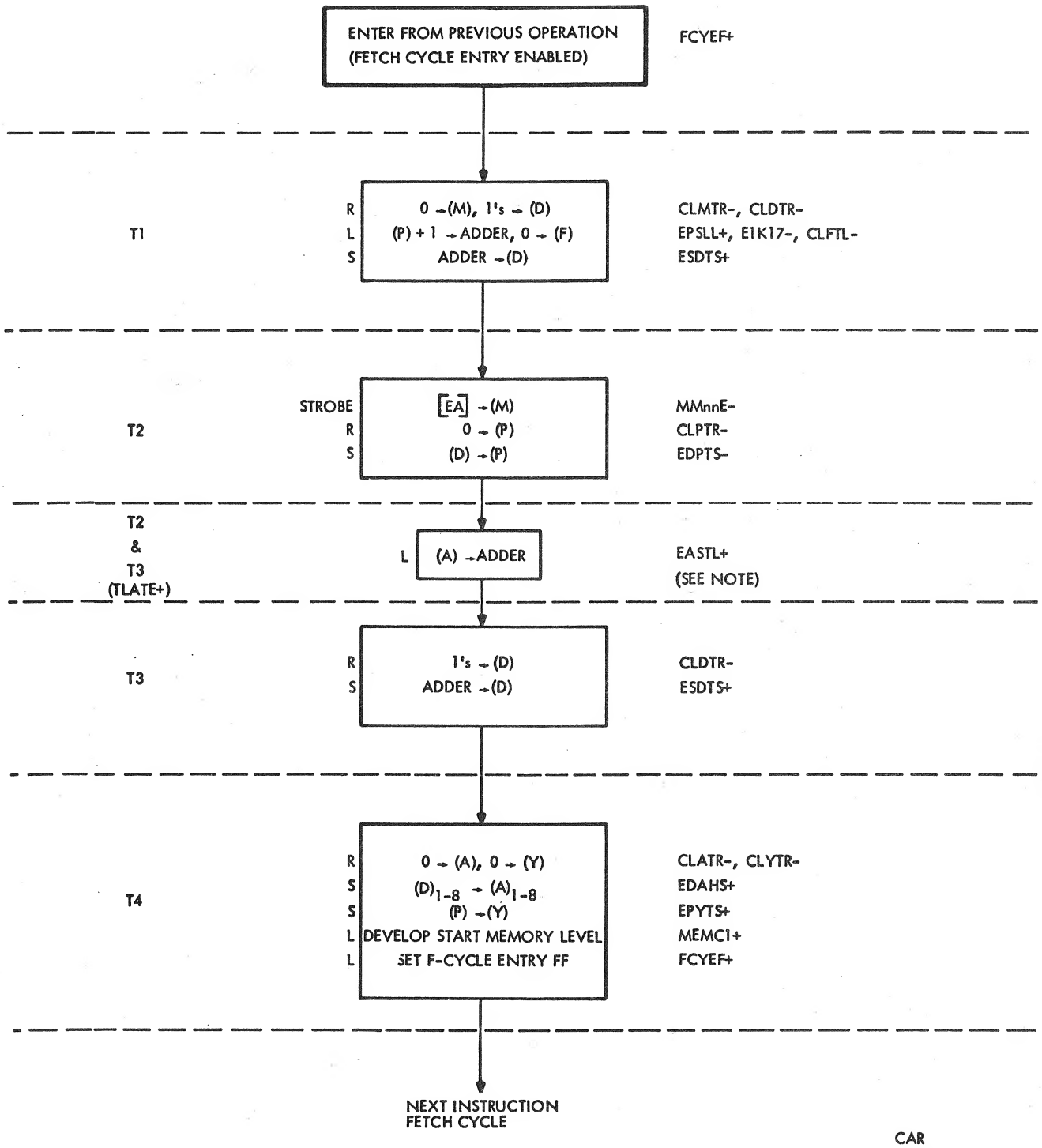
OP Code: 141050 Type: G, 1 cycle

Description: $0 \rightarrow (A)_{1-8}$




Execution Time ($\mu$sec): 0.96

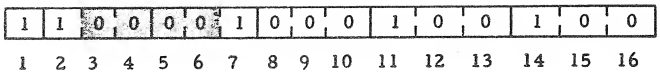
| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-J5 | 116-D7-D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L10 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (EDPTL+)(MCRST+)* | 129-H10 | 101--116-H10 | Clear P-register |
| EDPTS- | 129-L9 | F | TL2 | S | (EDPTL+)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register into P-register |
| EASTL+ | 127-L1 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+ | 129-G11 | 101--116-A5 | Enable A-register to adder |
| EMSHL+ | 127-L8 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--108--A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--108-A9 | Enable M-(1-7) to adder |
| EMSLL+ | 127-L10 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 109--116-A8 | Enable (M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 109--116-A9 | Enable M-(8-16) to adder |
| JAMKN- | 127-J3 | F | TLATE | L | (GEN0P+)(M01FF+)(M02FF+) | 127-C2 | 101--116-D6 | Force carry network to ZERO |
| CLDTR- | 127-J5 | F | TL3 | R | (TL3FF+)(ACYLF-) | 125-D6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLATR- | 122-H7 | F | TL4 | R | (GEN0A+)(M11FF+)(TL4FF+) | 122-D5 | 101--116-H5 | Clear A-register |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-D3 | 101--116-L11 | Clear Y-register |
| EDALS+ | 122-L2 | F | TL4 | S | (GEN0A+)(M11FF+)(M13FF+) | 129-D5/H2 | 109--116-G7 | Enable D-register to A(9-16) |
| EPYTS+ | 129-L4 | F | TL4 | S | (P1SEX-)(E01NS+)(0PGJS-) | 129-D4/H4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |


ENTER FROM PREVIOUS OPERATION
(FETCH CYCLE ENTRY ENABLED)
FCYEF+
T1
R
L
S
0 → (M), 1's → (D)
(P) + 1 → ADDER, 0 → (F)
ADDER → (D)
CLMTR-, CLDTR-
EPSLL+, E1K17-, CLFTL-
ESDTS+
T2
STROBE
R
S
[EA] → (M)
0 → (P)
(D) → (P)
MMnnE-
CLPTR-
EDPTS-
T2
&
T3
(TLATE+)
L
(A) → ADDER
EASTL+
(SEE NOTE)
T3
R
S
1's → (D)
ADDER → (D)
CLDTR-
ESDTS+
T4
R
S
S
L
L
0 → (A), 0 → (Y)
(D)1-8 → (A)1-8
(P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY FF
CLATR-, CLYTR-
EDAHS+
EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
CAR
1 CYCLE
OP CODE 141044
NOTE: MISSING SIGNALS CAN BE
FOUND IN CAR ANALYSIS
3400

Instruction: Clear Right Half (CAR)

OP Code: 141044 Type: G, 1 cycle

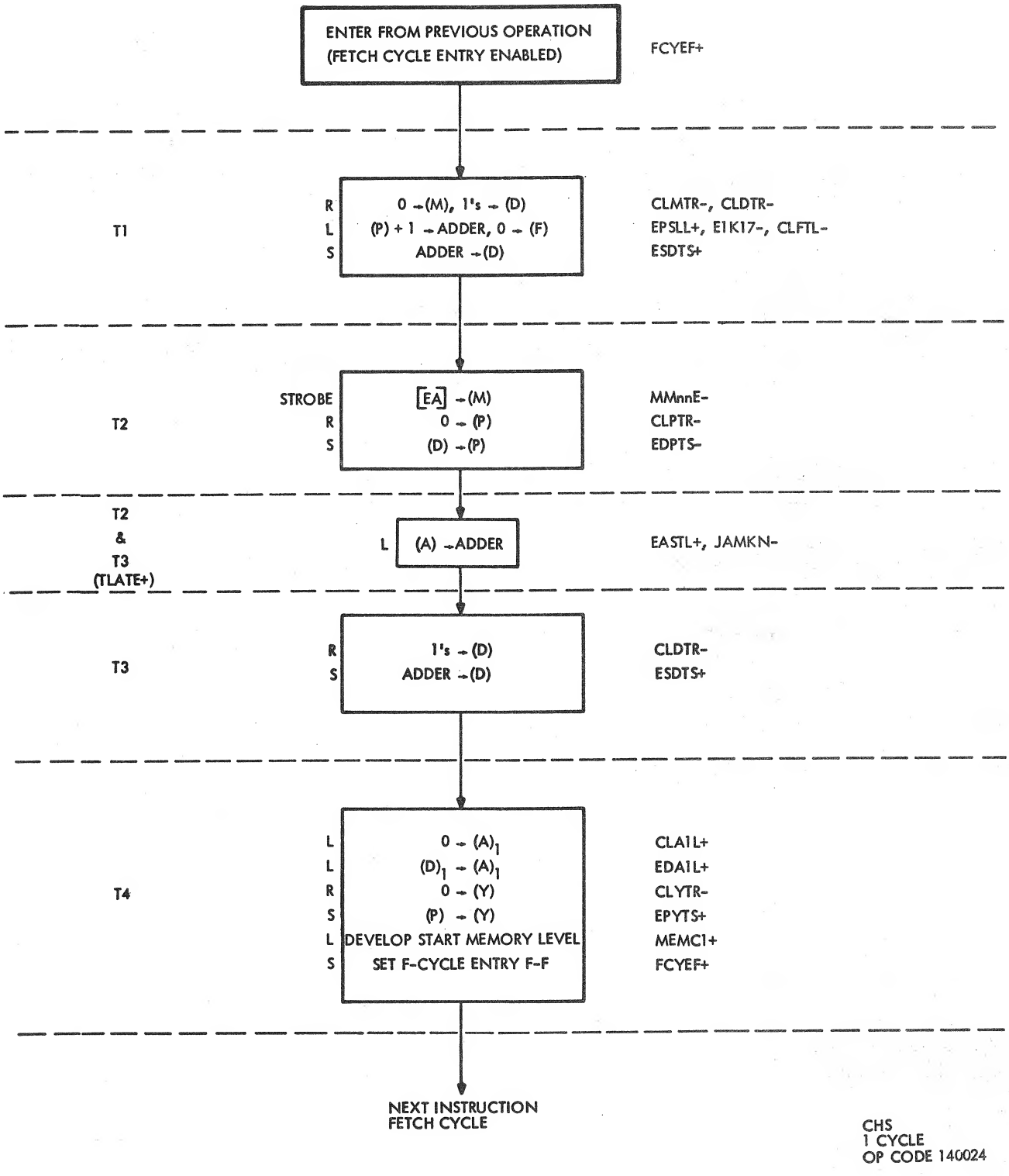
Description: $0 \rightarrow (A)_{9-16}$



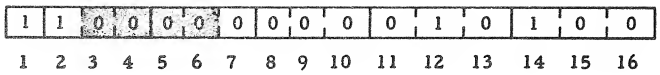

Execution Time ($\mu$sec): 0.96

| Function | Origin | Cyc | Tim | Clk | Boolean Expression | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-J5 | 116-D7-D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L10 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ FF |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (EDPTL+)(MCRST+)* | 129-H10 | 101--116-H10 | Clear P-register |
| EDPTS- | 129-L9 | F | TL2 | S | (EDPTL+)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register into P-register |
| EASTL+ | 127-L1 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 129-G11 | 101--116-A5 | Enable A-register to adder |
| EMSHL+ | 127-L8 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--108-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--108-A9 | Enable M-(1-7) to adder |
| EMSLL+ | 127-L10 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 109--116-A8 | Enable M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 109--116-A9 | Enable M-(8-16) to adder |
| JAMKN- | 127-J3 | F | TLATE | L | (GEN0P+)(M01FF+)(M02FF+) | 127-C2 | 101--116-D6 | Force carry network to ZERO |
| CLDTR- | 127-J5 | F | TL3 | R | (TL3FF+)(ACYLF-) | 125-D6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLATR- | 122-H7 | F | TL4 | R | (GEN0A+)(M11FF+)(TL4FF+) | 122-D5 | 101--116-H5 | Clear A-register |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-D3 | 101--116-L11 | Clear Y-register |
| EDAHS+ | 122-L1 | F | TL4 | S | (GEN0A+)(M11FF+)(M14FF+) | 122-D5/F2 | 101--108-G7 | Enable D-register to A-register 1-8 |
| EPYTS+ | 129-L4 | F | TL4 | S | (P1SEX-)(E01NS+)(0PGJS-) | 129-D4/H4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

*See gate A1A44-F at 129-D8 for EDPTL+

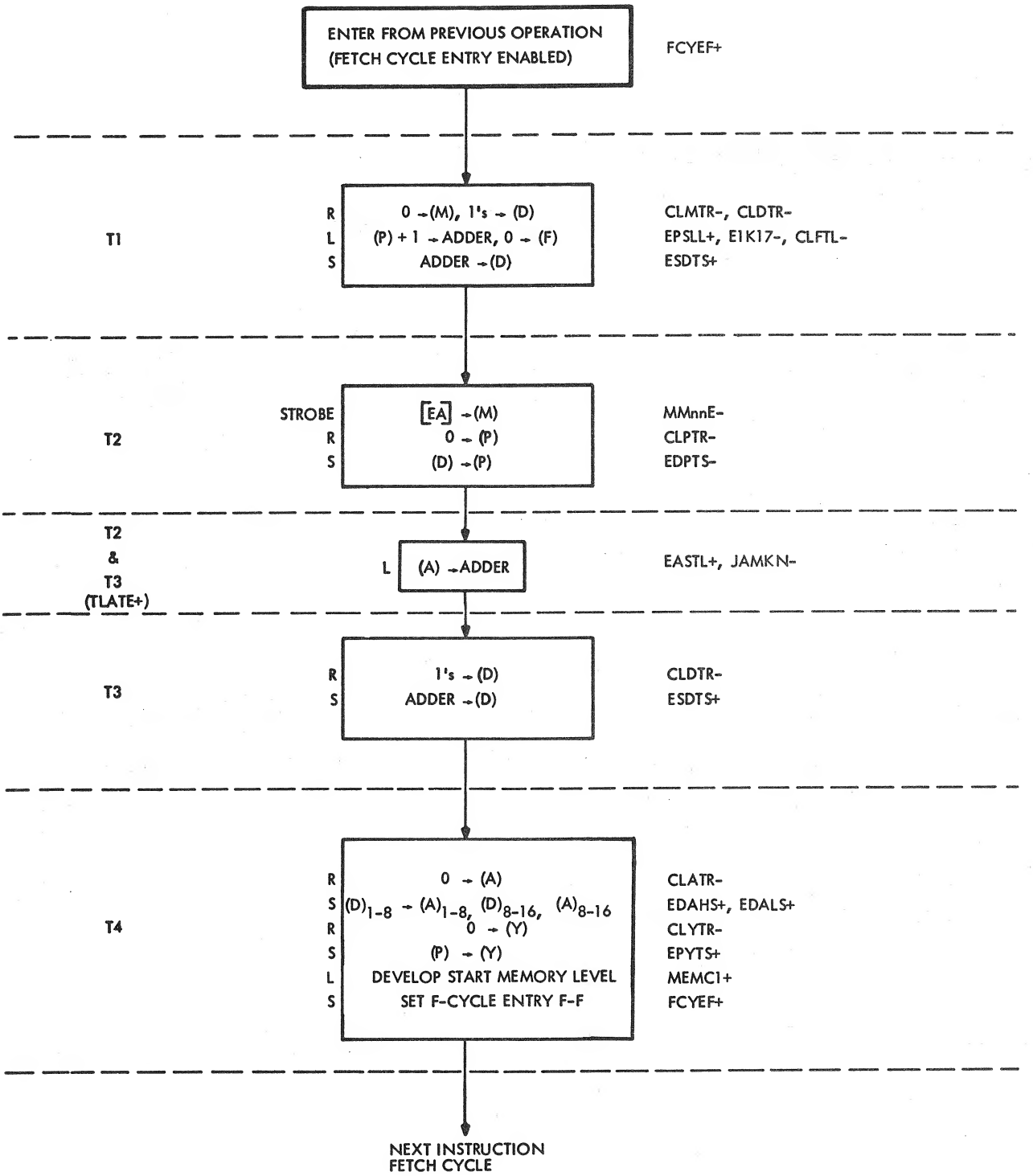

ENTER FROM PREVIOUS OPERATION
(FETCH CYCLE ENTRY ENABLED)
FCYEF+
T1
R
L
S
0 → (M), 1's → (D)
(P) + 1 → ADDER, 0 → (F)
ADDER → (D)
CLMTR-, CLDTR-
EPSLL+, E1K17-, CLFTL-
ESDTS+
T2
STROBE
R
S
[EA] → (M)
0 → (P)
(D) → (P)
MMnnE-
CLPTR-
EDPTS-
T2
&
T3
(TLATE+)
L
(A) → ADDER
EASTL+, JAMKN-
T3
R
S
1's → (D)
ADDER → (D)
CLDTR-
ESDTS+
T4
L
L
R
S
L
S
0 → (A)1
(D)1 → (A)1
0 → (Y)
(P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY F-F
CLA1L+
EDA1L+
CLYTR-
EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
CHS
1 CYCLE
OP CODE 140024
5480

Instruction: Complement A Sign (CHS)
OP Code: 140024 Type: G, 1 cycle
Description: ONE's complement of $(A_1) \rightarrow (A_1)$




Execution Time ($\mu$sec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-C3/ J6 | 116-D7/D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCRST+) | 129-D7/ H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| EASTL+ | 127-L1 | F | TLATE | L | See gate A1E37-D | 127-C2 | 101--116-A5 | Enable A-register to adder |
| JAMKN- | 127-J3 | F | TLATE | L | See gate A1E37-D | 127-C2 | 101--116-D6<br>117-B/D | Jam carry network |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLA1L+ | 130-H11 | F | TL4 | L | (GEN0A+)(TL4FF+)(M14FF+) | 130-F9 | 101-H3 | Clear A-register bit 1 |
| EDA1L+ | 130-H10 | F | TL4 | L | (GEN0A+)(TL4FF+)(M14FF+) | 130-F9 | 101-H6 | Enable D-register bit 1 into A-register bit 1 |
| CLYTR- | 129-J3 | F | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-D2 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/ H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

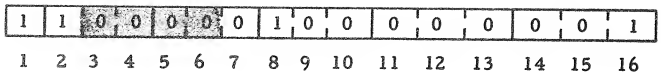



CMA
1 CYCLE
OP CODE 1401401

Instruction: Complement A (CMA)

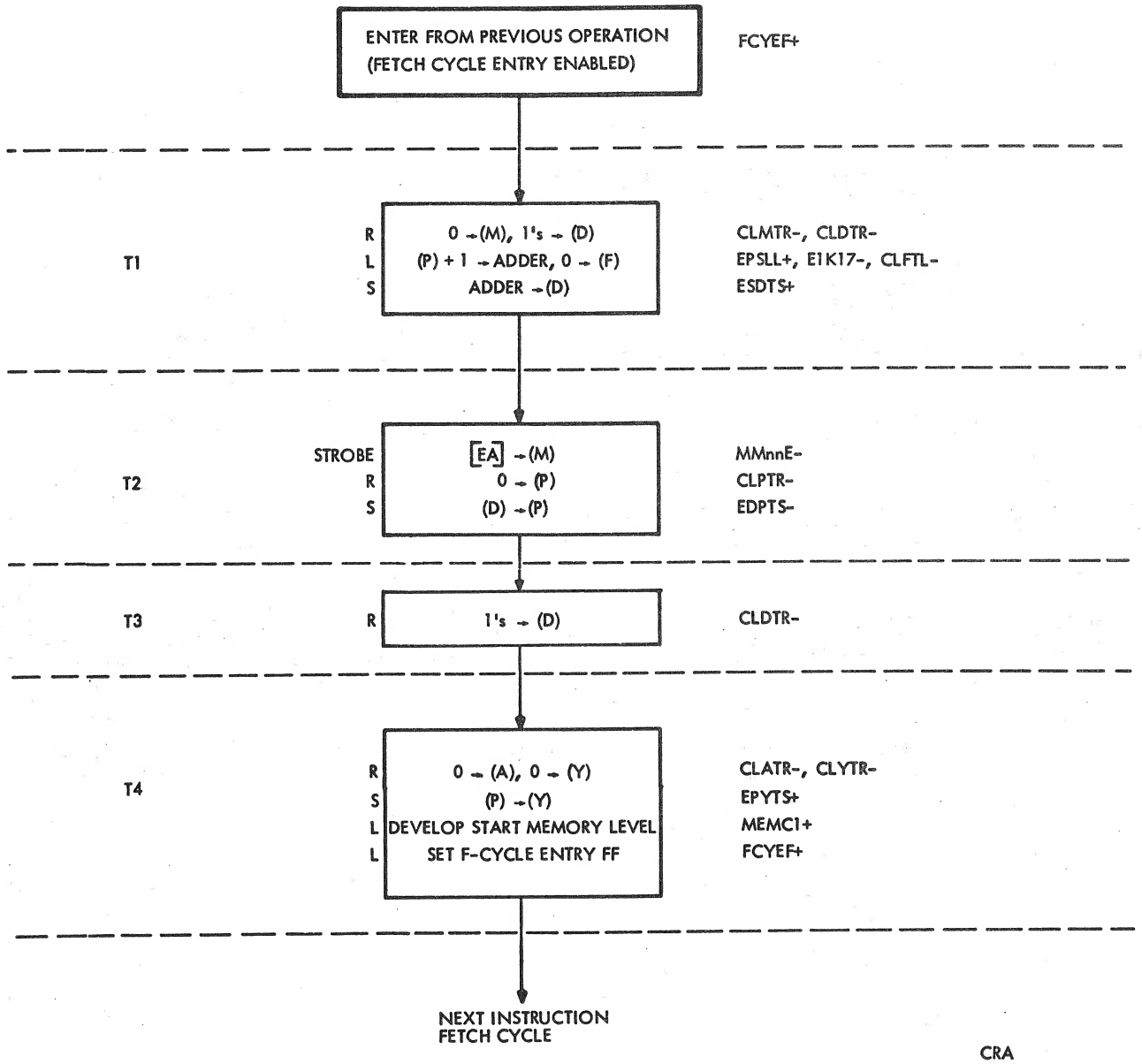
OP Code: 140401 Type: G, 1 cycle

Description: ONE's complement of (A) → (A)




Execution Time (μsec): 0.96

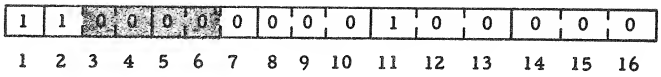
| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-C3/ J6 | 116-D7-D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCRST+) | 128-D7/ H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| EASTL+ | 127-L1 | F | TLATE | L | See gate A1E37-D | 127-C2 | 101--116-A5 | Enable A-register to adder |
| JAMKN- | 127-J3 | F | TLATE | L | See gate A1E37-D | 127-C2 | 101--116-D6<br>117 B/D | Jam carry network |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLATR- | 122-H7 | F | TL4 | R | (M16FF+)(GEN0A+)(TL4FF+)(MCRST+) | 122-A3 | 101--116-H5 | Clear A-register |
| EDAHS+ | 122-L1 | F | TL4 | S | (M16FF+)(GEN0A+)(TL4FF+)(MCSET+) | 122-A3 | 101--108-G7 | Enable D-register to A(1-8) |
| EDALS+ | 122-L2 | F | TL4 | S | (M16FF+)(GEN0A+)(TL4FF+)(MCSET+) | 122-A3 | 108--116-G7 | Enable D-register to A(9-16) |
| CLYTR- | 129-J3 | F | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-D2 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11 H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |


ENTER FROM PREVIOUS OPERATION
(FETCH CYCLE ENTRY ENABLED)
FCYEF+
T1
R
L
S
0 → (M), 1's → (D)
(P) + 1 → ADDER, 0 → (F)
ADDER → (D)
CLMTR-, CLDTR-
EPSLL+, E1K17-, CLFTL-
ESDTS+
T2
STROBE
R
S
[EA] → (M)
0 → (P)
(D) → (P)
MMnnE-
CLPTR-
EDPTS-
T3
R
1's → (D)
CLDTR-
T4
R
S
L
L
0 → (A), 0 → (Y)
(P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY FF
CLATR-, CLYTR-
EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
CRA
1 CYCLE
OP CODE 140040
NOTE: MISSING SIGNALS CAN BE
FOUND IN CRA ANALYSIS.
3390

Instruction: Clear A-register (CRA)

OP Code: 140040 Type: Generic, 1 cycle

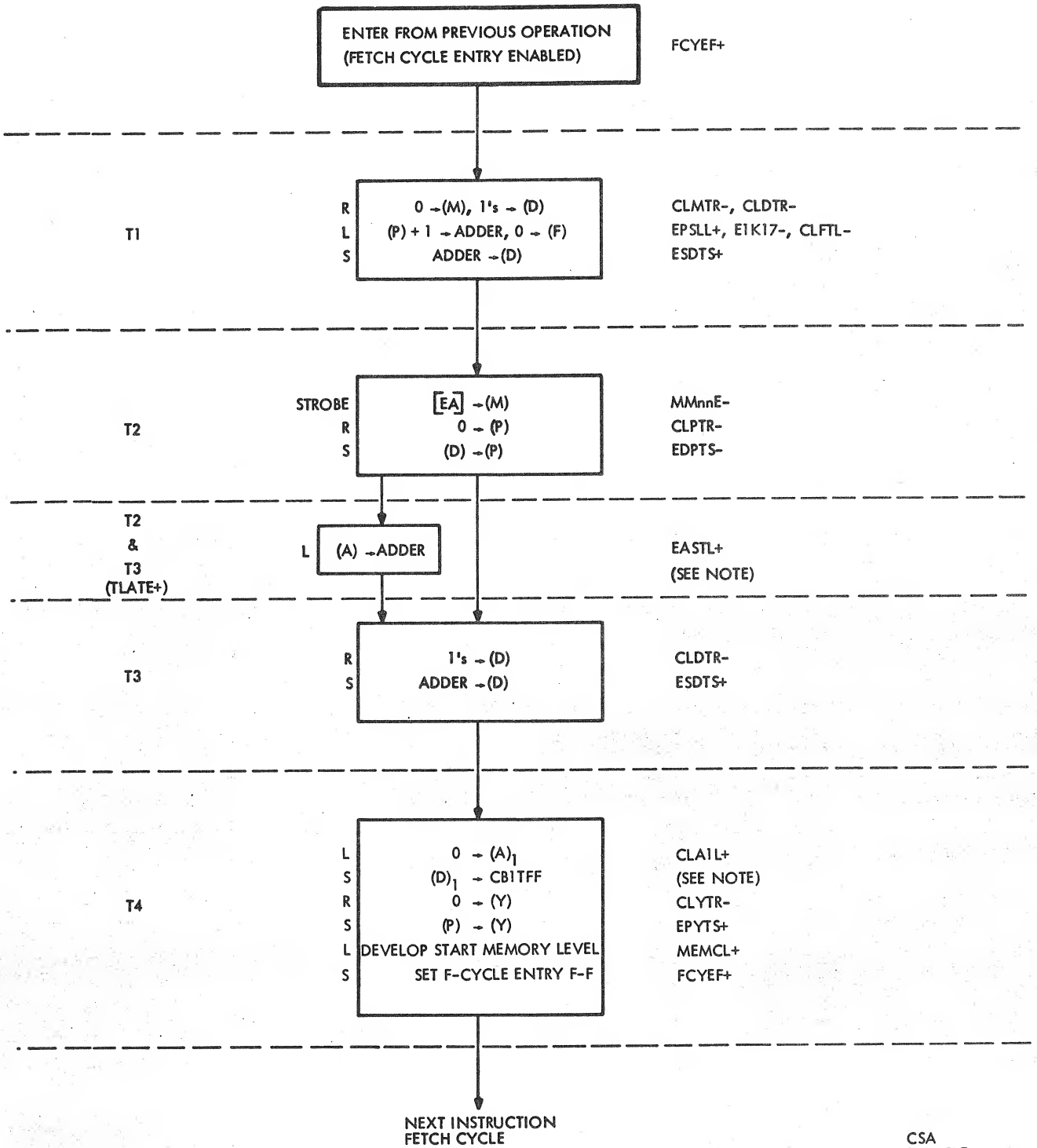
Description: $0s \rightarrow (A)_{1-16}$




Execution Time ($\mu$sec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17+ | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-J5 | 116-D7/D9 | Force carry-in to bit 16 |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Reset M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1 | Reset F-register |
| | | | | | | | 121-A5 | Reset shift counter |
| | | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 125-L10 | Reset AZZZZ+ |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4/D6/D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (EDPTL+)* | 129-F9 | 101--116-H10 | Reset P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (EDPTL+)* | 129-F9 | 101--116-G11 | Enable D-register into P-register |
| CLDTR- | 125-J6 | F | TL3 | R | (TL3FF+)(ACYLF-) | 125-D6 | 101--116-E7 | Clear D-register to ONEs |
| CLATR- | 122-J7 | F | TL4 | R | (CLATL+)(MCRST+)** | 122-F7 | 101--116-H5 | Reset A-register |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-D3 | 101--116-L11 | Reset Y-register |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-) | 129-D4 | 101--116-J11 | Enable P-register into Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (MEMC1-)*** | 126-G11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | XXXX | Start memory cycle |
| FCYEF+ | 119-F10 | F | TL4 | L | (Set: (TL4FF+)(E01NS+) | 119-D10 | 119-H9 | Enable set FCYLF+ at next TL1FF+ |

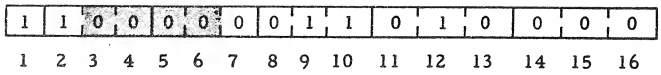
*(EDPTL+) @ 129-F9 = (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-) @ 129-D8
**(CLATL+) @ 122-F7 = (GEN0A+)(M11FF+)(TL4FF+) @ 122-C5
***(MEMC1-) @ 126-G11 = (TL4FF+)(SPM0D-)((10GRP+)(FCYLF+)-)


ENTER FROM PREVIOUS OPERATION
(FETCH CYCLE ENTRY ENABLED)
FCYEF+
T1
R
L
S
0 → (M), 1's → (D)
(P) + 1 → ADDER, 0 → (F)
ADDER → (D)
CLMTR-, CLDTR-
EPSLL+, E1K17-, CLFTL-
ESDTS+
T2
STROBE
R
S
[EA] → (M)
0 → (P)
(D) → (P)
MMnnE-
CLPTR-
EDPTS-
T2
&
T3
(TLATE+)
L
(A) → ADDER
EASTL+
(SEE NOTE)
T3
R
S
1's → (D)
ADDER → (D)
CLDTR-
ESDTS+
T4
L
S
R
S
L
S
0 → (A)1
(D)1 → CB1TFF
0 → (Y)
(P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY F-F
CLA1L+
(SEE NOTE)
CLYTR-
EPYTS+
MEMCL+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
CSA
1 CYCLE
OP CODE 140320
NOTE: MISSING SIGNALS CAN BE
FOUND IN CSA ANALYSIS
3481

Instruction: Copy Sign and Set Sign Plus (CSA)

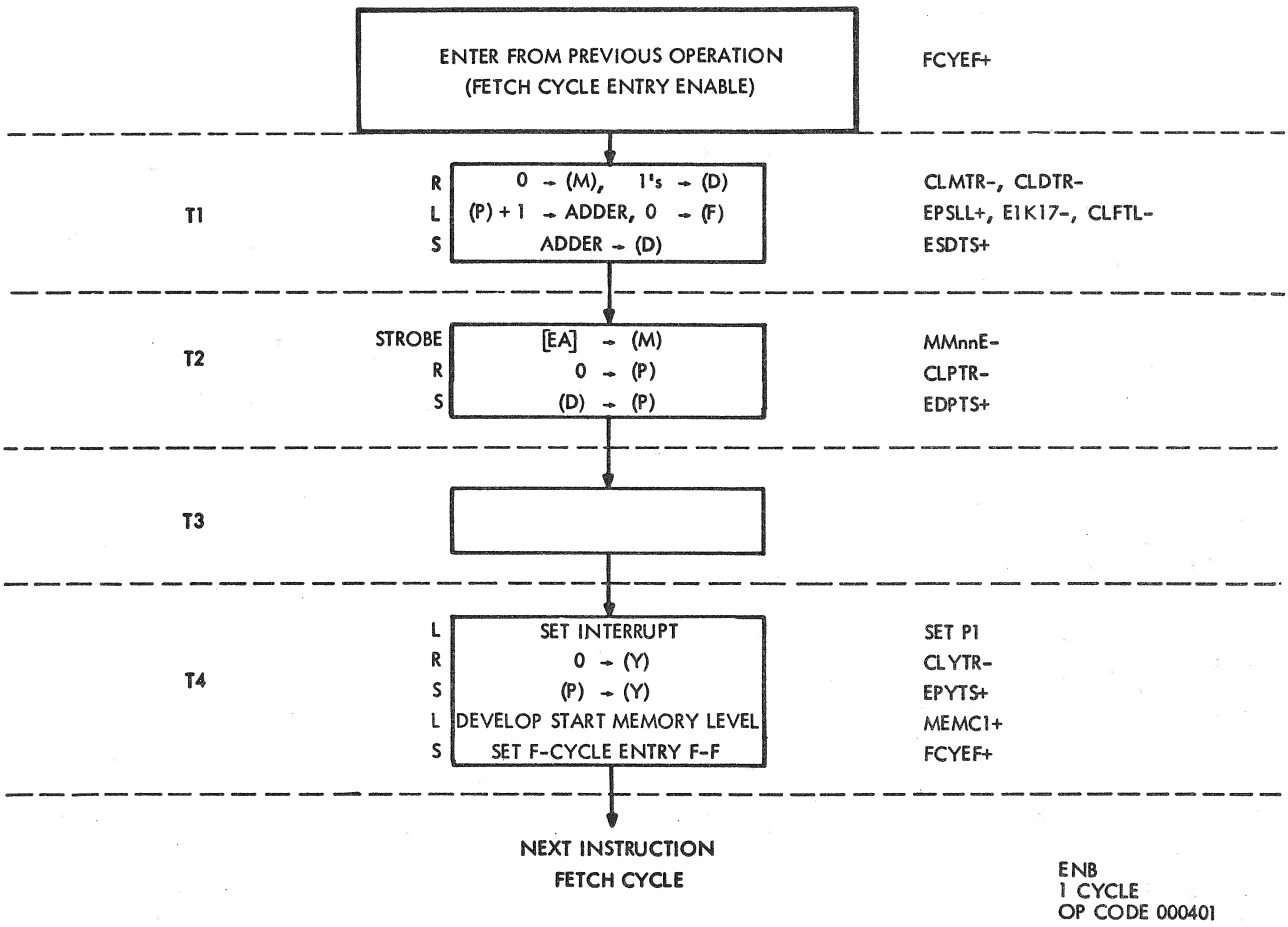
OP Code: 140320 Type: G, 1 cycle

Description: $(A)_1 \rightarrow C$, $0 \rightarrow (A)_1$




Execution Time ($\mu$sec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-C3/ J6 | 116-D7-D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF+)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+) (MEMAC-)(MCRST+) | 129-D7/ H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+) (MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| EASTL+ | 127-L1 | F | TLATE | L | (See gate A1E37-DV gate A1C43A) | 127-C2 | 101--116-A5 117 B/D | Enable A-register to adder |
| JAMKN- | 127-J3 | F | TLATE | L | (See gate A1E37-DV gate A1C43A) | 127-C2 | 101--116-D6 | Jam carry network |
| EMSHL+ | 127-L8 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 101--107-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 101--107-A9 | Enable M-(1-7)to adder |
| EMSLL+ | 127-L10 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 108--116-A8 | Enable M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G1 | 108--116-A9 | Enable M-(8-16)to adder |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLA1L+ | 130-H11 | F | TL4 | L | (GEN0A+)(TL4FF+)(M10FF+) | 130-F11 | 101-H3 | Clear A-register bit 1 |
| D1 → C | 124-L2 | F | TL4 | S | (GEN0A+)(TL4FF+)(M10FF+) (M12FF+)(D01FF+) | 124-D9 | 124-L2 | Set D1 into C-bit |
| CLYTR- | 129-J3 | F | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-D2 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+) (0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11 H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

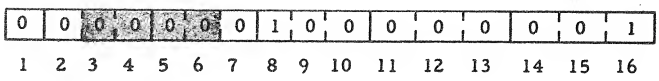

ENTER FROM PREVIOUS OPERATION
(FETCH CYCLE ENTRY ENABLE)
FCYEF+
T1
R
L
S
0 → (M), 1's → (D)
(P) + 1 → ADDER, 0 → (F)
ADDER → (D)
CLMTR-, CLDTR-
EPSLL+, E1K17-, CLFTL-
ESDTS+
T2
STROBE
R
S
[EA] → (M)
0 → (P)
(D) → (P)
MMnnE-
CLPTR-
EDPTS+
T3
T4
L
R
S
L
S
SET INTERRUPT
0 → (Y)
(P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY F-F
SET P1
CLYTR-
EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
ENB
1 CYCLE
OP CODE 000401

5499

Instruction: Enable Program Interrupt (ENB)

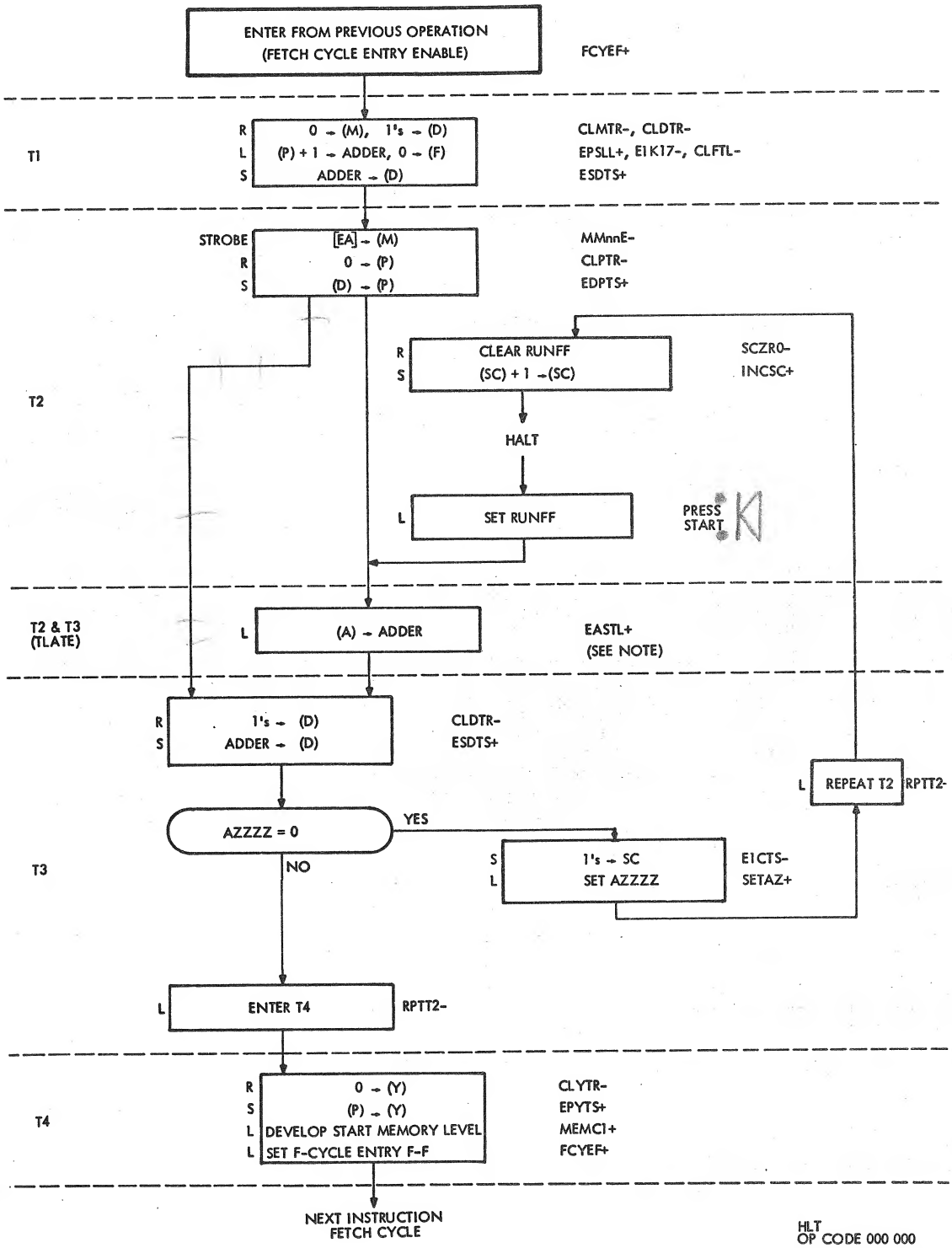
OP Code: 000401 Type: G, 1 cycle

Description: Set machine status to permit interrupt




Execution Time ($\mu$sec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-C3/ J6 | 116-D7-D9 | Force carrv to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCRST+) | 129-D7/ H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| SET P1 | 134-A4 | F | TL4 | L | (GEN0B+)(TL4FF+)(M08FF+) | 134-A4 | 134-C4 | Permit interrupt |
| CLYTR- | 129-J3 | F | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-D2 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/ H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |



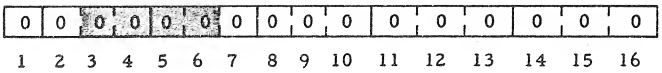

NOTE: MISSING SIGNALS CAN BE FOUND IN HLT ANALYSIS

3489

Instruction: Halt (HLT)

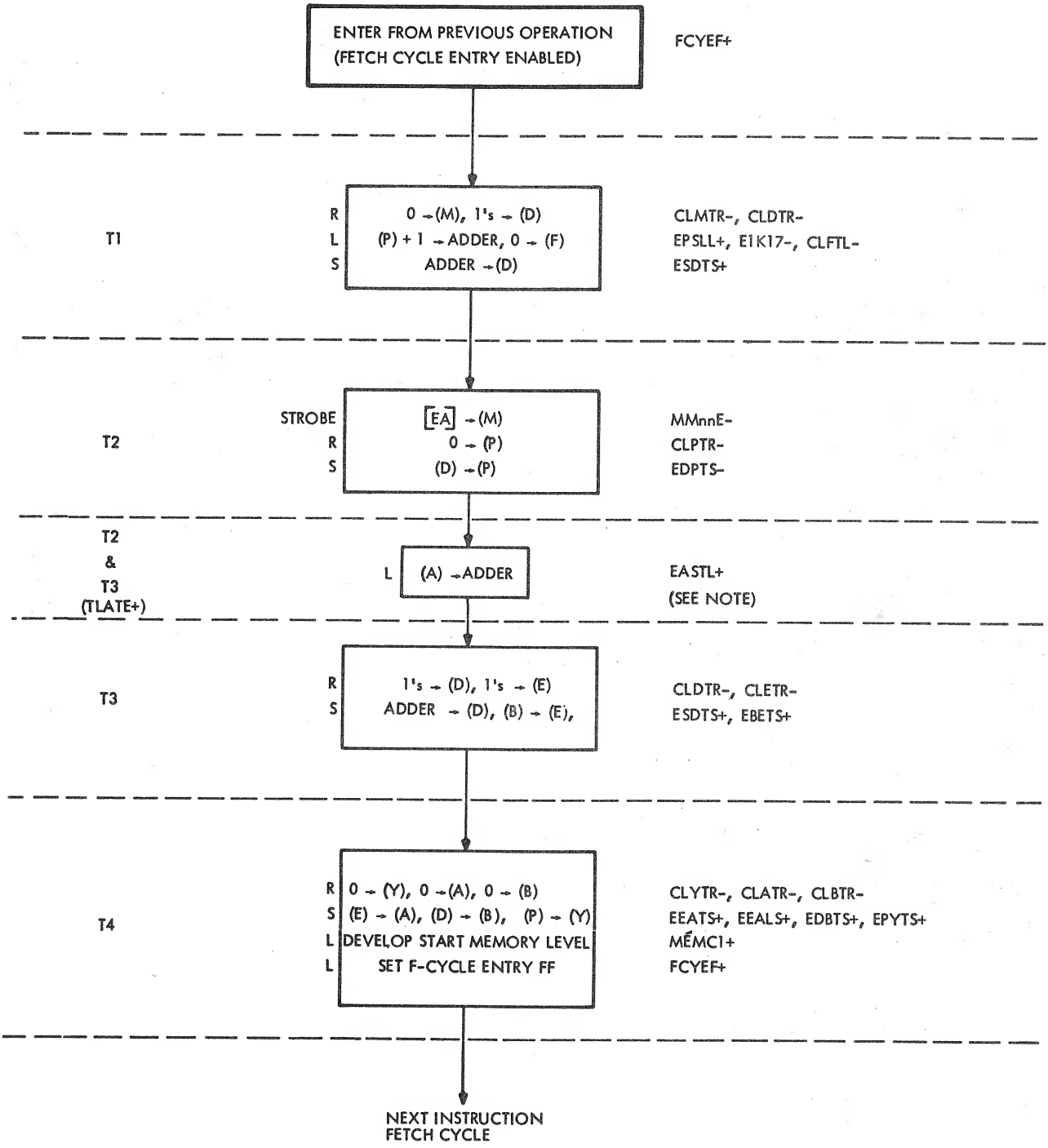
OP Code: 000000 Type: G, 1.5 cycles

Description: Set machine to halt mode




Execution Time ($\mu$sec): 1.44

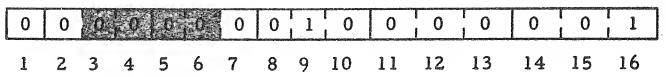
| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-J5 | 116-D7/D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 121-A5<br>120-B1<br>125-L10 | Clear shift counter<br>Clear F-register<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-H10 | F | TL2 | R | (EDPTL+)(MCRST+) | 129-D7/F9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (EDPTL+)(MCSET+) | 129-D7/H9 | 101--116-G11 | Enable D-register to P-register |
| EASTL+ | 127-L1 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G11 | 101--116-A5 | Enable A-register to adder |
| EMSHL+ | 127-L8 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G11 | 101--107-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G11 | 101--107-A9 | Enable M-(1-7) to adder |
| EMSHL+ | 127-L10 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G11 | 108--116-A8 | Enable M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G11 | 108--116-A9 | Enable M-(8-16) to adder |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| E1CTS- | 125-H8 | F | TL3 | S | (GEN0B+)(TL3FF+)(M16FF-)(MCSET+)(AZZZZ-) | 125-D8/H8 | 121-A8<br>126-F5 | Set shift counter to all ONEs<br>Repeat TL2 (RPTT2+) |
| SETAZ+ | 125-J9 | F | TL3 | L | (GEN0B+)(TL3FF+)(M16FF-) | 125-D8 | 125-H9 | Set AZZZZ F-F |
| RUNFF- | 126-H2 | F | TL2 | R | (MCRST+)(TL2FF+)(GEN0B+)(M16FF-)(SCZR0-)(RESTR-) | 126-F3 | 126-H1 | Reset RUNFF |
| INCSC- | 126-L3 | F | TL2 | L | (FCYEF+)(TL2FF+) | 126-H3 | 121-A5 | Enable increment shift counter |
| | | | | | Depress and release START button | | | |
| RUNFF+ | 126-H2 | | | | (READYFF+)(P1L00-) | 126-D3 | 126-H2 | Set RUNFF |
| | | | | | Repeat TLATE and TL3 with entry to TL4 (SCZR0+)(RPTT2-) | | | |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-D3, | 101--116-L11 | Clear Y-register |
| EPYTS+ | 129-L4 | F | TL4 | S | (P1SEX-)(E01NS+)(0PGJS-) | 129-D4/H4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |


ENTER FROM PREVIOUS OPERATION
(FETCH CYCLE ENTRY ENABLED)
FCYEF+
T1
R
L
S
0 → (M), 1's → (D)
(P) + 1 → ADDER, 0 → (F)
ADDER → (D)
CLMTR-, CLDTR-
EPSLL+, E1K17-, CLFTL-
ESDTS+
T2
STROBE
R
S
[EA] → (M)
0 → (P)
(D) → (P)
MMnnE-
CLPTR-
EDPTS-
T2
&
T3
(TLATE+)
L
(A) → ADDER
EASTL+
(SEE NOTE)
T3
R
S
1's → (D), 1's → (E)
ADDER → (D), (B) → (E),
CLDTR-, CLETR-
ESDTS+, EBETS+
T4
R
S
L
L
0 → (Y), 0 → (A), 0 → (B)
(E) → (A), (D) → (B), (P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY FF
CLYTR-, CLATR-, CLBTR-
EEATS+, EEALS+, EDBTS+, EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
NOTE: MISSING SIGNALS CAN BE
FOUND IN IAB ANALYSIS
5391
IAB
1 CYCLE
OP CODE 000201

Instruction: Interchange A and B (IAB)

OP Code: 000201 Type: Generic, 1 cycle

Description: (A) ⇄ (B)

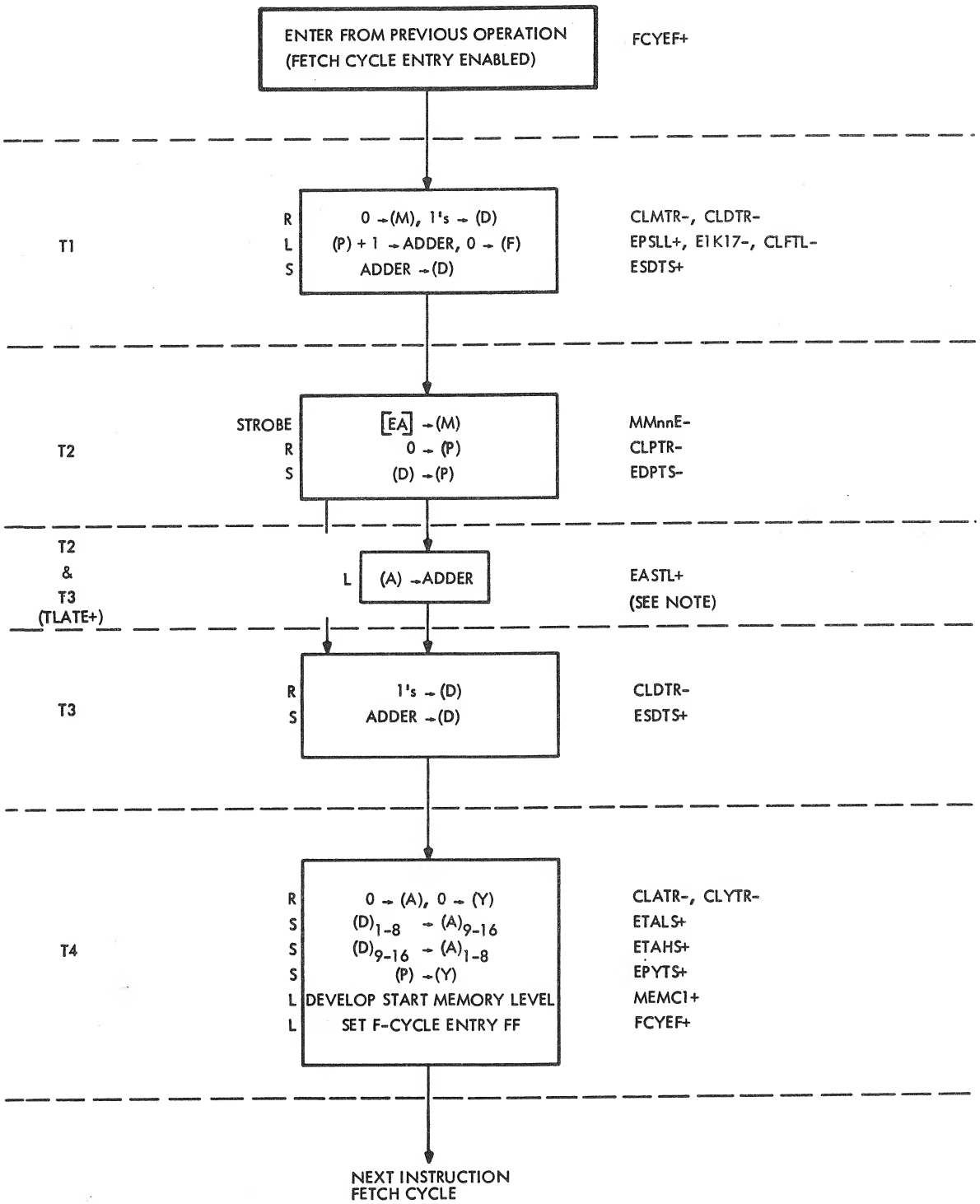



Execution Time (μsec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | TLATE | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | TLATE | L | (TLATE-) | 127-J5 | 116-D7/D9 | Force carry-in to bit 16 |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Reset M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L10 | Reset F-register<br>Reset shift counter<br>Reset AZZZZ+ F-F |
| ESDTS+ | 125-L4 | F | TLATE | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4/D6/D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (EDPTL+)* | 129-F9 | 101--116-H10 | Reset P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (EDPTL+)* | 129-F9 | 101--116-G11 | Enable D-register into P-register |
| EASTL+ | 127-L1 | F | TLATE* | | (GEN0P+)(TLATE+)(M01FF-) | 127-G10 | 101--116-A5 | Enable A-register to adder |
| EMSLL+ | 127-L10 | F | TLATE* | | (GEN0P+)(TLATE+)(M01FF-) | 127-G10 | 108--116-A8 | Enable M(8-16) to adder |
| EMSHL+ | 127-L8 | F | TLATE* | | (GEN0P+)(TLATE+)(M01FF-) | 127-G10 | 101--107-A8 | Enable M(1-7) to adder |
| ENSLL+ | 127-L5 | F | TLATE* | | (GEN0P+)(TLATE+)(M01FF-) | 127-G10 | 108--116-A9 | Enable M-(8-16) to adder |
| ENSHL+ | 127-L7 | F | TLATE* | | (GEN0P+)(TLATE+)(M01FF-) | 127-G10 | 101--107-A9 | Enable M-(1-7) to adder |
| CLDTR- | 125-J6 | F | TL3 | R | (TL3FF+)(ACYLF-) | 125-D6 | 101--116-E7 | Clear D-register to ONEs |
| CLETR- | 125-J2 | F | TL3 | R | (TL3FF+)(GEN0P+)(M01FF-)(M09FF+) | 125-D1 | 101--116-K3 | Reset E-register |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-) | 125-D6 | 101--116-D4/D6/D8 | Enable adder sum into D-register |
| EBETS+ | 125-L1 | F | TL3 | S | (TL3FF+)(GEN0P+)(M01FF-)(M09FF+) @ 125-A1] | 125-D1 | 101--116-J2 | Enable B-register into E-register |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-D3 | 101--116-L11 | Reset Y-register |
| CLATR- | 122-H7 | F | TL4 | R | (CLATL+)* | 122-F7 | 101--116-H5 | Reset A-register |
| CLBTR- | 123-J6 | F | TL4 | R | (M5G4G-)* | 123-E2 | 101--116-H2 | Reset B-register |
| EEATS+ | 122-L3 | F | TL4 | S | (M5G4G-)* | 122-F3 | 101--110-G4 | Enable E(1-10) into A(1-10) |
| EEALS+ | 122-J4 | F | TL4 | S | (M5G4G-)* | 122-F3 | 101--116-G4 | Enable E(11-16) into A(11-16) |
| EDBTS+ | 123-L2 | F | TL4 | S | (M5G4G-)* | 123-E2 | 101--116-G3 | Enable D-register into B-register |
| FCYEF+ | 119-F10 | F | TL4 | L | (Set: (TL4FF+)(E01NS+)) | 119-D10 | 119-H9 | Enable set FCYLF+ at next TL1FF+ |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-) | 129-D4 | 101--116-J11 | Enable P-register into Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (MEMC1-) | 126-G11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | XXXX | Start memory cycle |

*EDPTL+ @ 129-F9 = (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-) @ 129-D8
*TLATE+ = ((TL2FF-)∨(TL3FF-)) @ 118-B7

*CLATL+ @ 122-F7 = (M5G4G-) @ 122-E7 = (GEN0B+)(TL4FF+)(M09FF+) @ 123-E2




NOTE: MISSING SIGNALS CAN BE FOUND IN ICA ANALYSIS

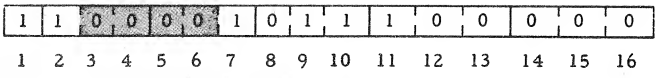
ICA
1 CYCLE
OP CODE 141340

3398

Instruction: Interchange Halves (ICA)

OP Code: 141340 Type: G, 1 cycle

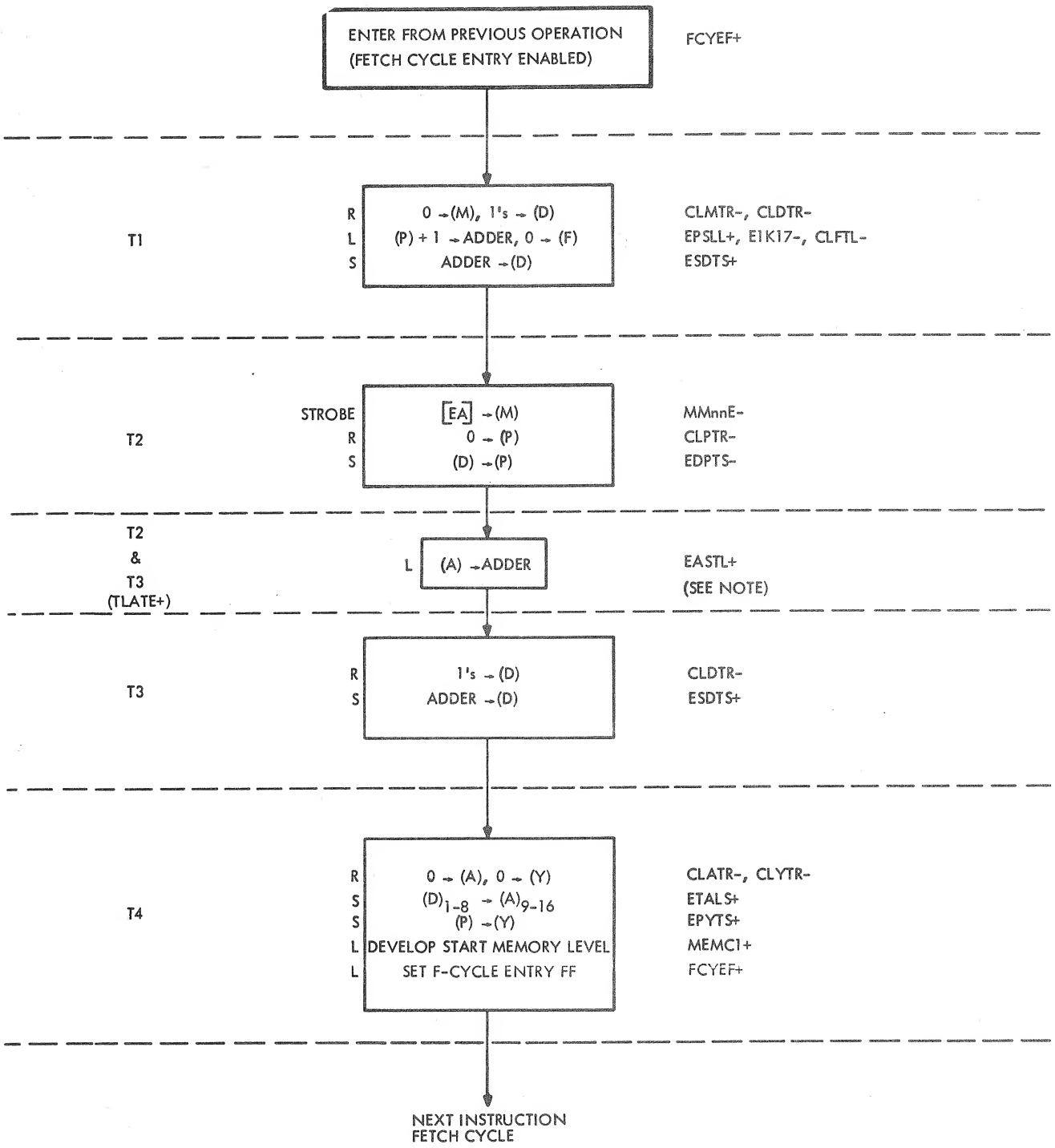
Description: $(A)_{1-8} \rightarrow (A)_{9-16}$
$(A)_{9-16} \rightarrow (A)_{1-8}$




Execution Time ($\mu$sec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-J5 | 116-D7/D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| JAMKN- | 127-J5 | F | TL1 | L | [(TLATE+)(ACYEF+)]- | 127-C3 | 127-L4 | Implement E1K17- |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 121-A5<br>120-B1<br>125-L10 | Clear shift counter<br>Clear F-register<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (EDPTL+)(MCRST+)* | 129-H9 | 101--116-H10 | Clear P-register |
| EDPTS- | 129-L9 | F | TL2 | S | (EDPTL+)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register into P-register |
| EASTL+ | 127-L1 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 129-G11 | 101--116-A5 | Enable A-register to adder |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-) | 125-D6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| EMSHL+ | 127-L8 | F | TLATE+ | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--108-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE+ | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--108-A9 | Enable M-(1-7) to adder |
| EMSLL+ | 127-L10 | F | TLATE+ | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 109--116-A8 | Enable M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE+ | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 109-116-A9 | Enable M-(8-16) to adder |
| JAMKN- | 127-J3 | F | TLATE+ | L | (GEN0P+)(M01FF+)(M02FF+) | 127-C2 | 101--116-D6 | Force carry network to ZERO |
| CLATR- | 122-H7 | F | TL4 | R | (GEN0A+)(M11F+)(TL4FF+) | 122-D5 | 101--116-H5 | Clear A-register |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-D3 | 101--116-L11 | Clear Y-register |
| ETALS+ | 122-K6 | F | TL4 | S | (GEN0A+)(M10FF+)(M11FF+) | 122-D5/H6 | 109-116-D4 | Enable D(1-8) into A(9-16) |
| ETAHS+ | 122-L5 | F | TL4 | S | (GEN0A+)(M09FF+)(M11FF+) | 122-D5/H5 | 101--108-D4 | Enable D(9-16) into A(1-8) |
| EPYTS+ | 129-L4 | F | TL4 | S | (P1SEX-)(E01NS+)(0PGJS-) | 129-D4/H4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

*See gate A1A44-F at 129-D8 for EDPTL+




NOTE: MISSING SIGNALS CAN BE FOUND IN ICL ANALYSIS

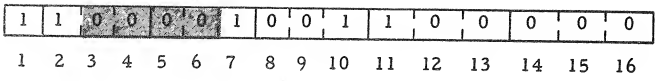
ICL
1 CYCLE
OP CODE 141140

3397

Instruction: Interchange and Clear Left Half (ICL)

OP Code: 141140 Type: G, 1 cycle

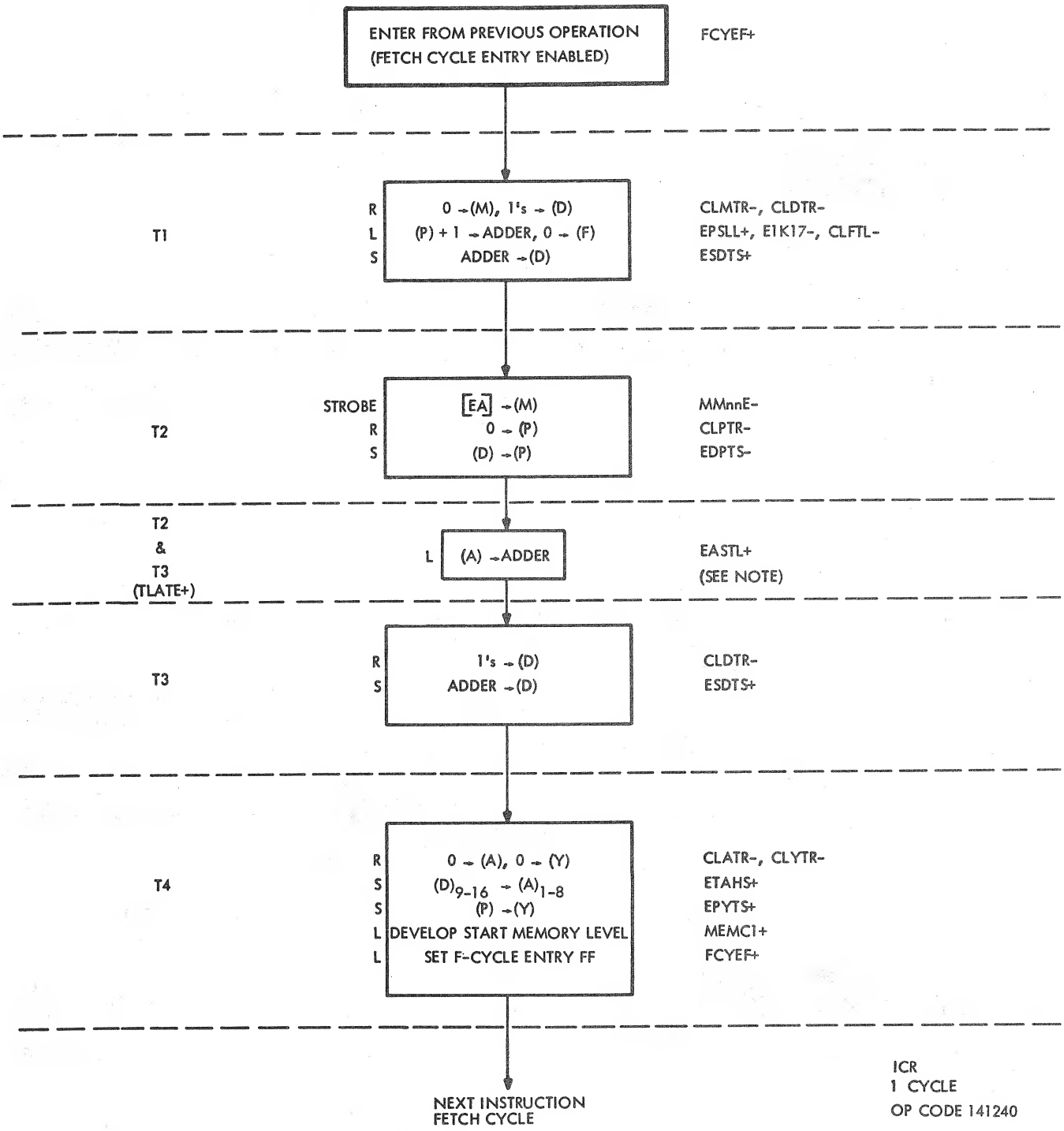
Description: $(A)_{1-8} \rightarrow (A)_{9-16}$
$0 \rightarrow (A)_{1-8}$




Execution Time ($\mu$sec): 0.96

| Function | Origin | Cyc | Tim | Clk | Boolean Expression | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-J5 | 116-D7-D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L10 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ FF |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (EDPTL+)(MCRST+)* | 129-H10 | 101--116-H10 | Clear P-register |
| EDPTS- | 129-L9 | F | TL2 | S | (EDPTL+)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register into P-register |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| EMSHL+ | 127-L8 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--108-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 101--108-A9 | Enable M-(1-7) to adder |
| EMSLL+ | 127-L10 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 109--116-A8 | Enable M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 127-G11 | 109--116-A9 | Enable M-(8-16) to adder |
| JAMKN- | 127-J3 | F | TLATE | L | (GEN0P+)(M01FF+)(M02FF+) | 127-C2 | 101--116-D6 | Force carry network to ZERO |
| EASTL+ | 127-L1 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) | 129-G11 | 101--116-A5 | Enable A-register to adder |
| CLDTR- | 127-J5 | F | TL3 | R | (TL3FF+)(ACYLF-) | 125-D6 | 101--116-E7 | Clear D-register to ONEs |
| CLATR- | 122-H7 | F | TL4 | R | (GEN0A+)(M11FF+)(TL4FF+) | 122-D5 | 101--116-H5 | Clear A-register |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-D3 | 101--116-L11 | Clear Y-register |
| ETALS+ | 122-K6 | F | TL4 | S | (GEN0A+)(M10FF+)(M11FF+) | 122-D5/H6 | 109--116-D4 | Enable D-register (1-8) into A-register (9-16) |
| EPYTS+ | 129-L4 | F | TL4 | S | (P1SEX-)(E01NS+)(0PGJS-) | 129-D4/H4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

*See gate A1A44-F at 129-D8 for EDPTL+



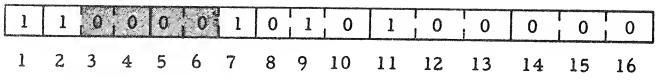

NOTE: MISSING SIGNALS CAN BE FOUND IN ICR ANALYSIS

3596

Instruction: Interchange and Clear Right Half (ICR)

OP Code: 141240 Type: G, 1 cycle

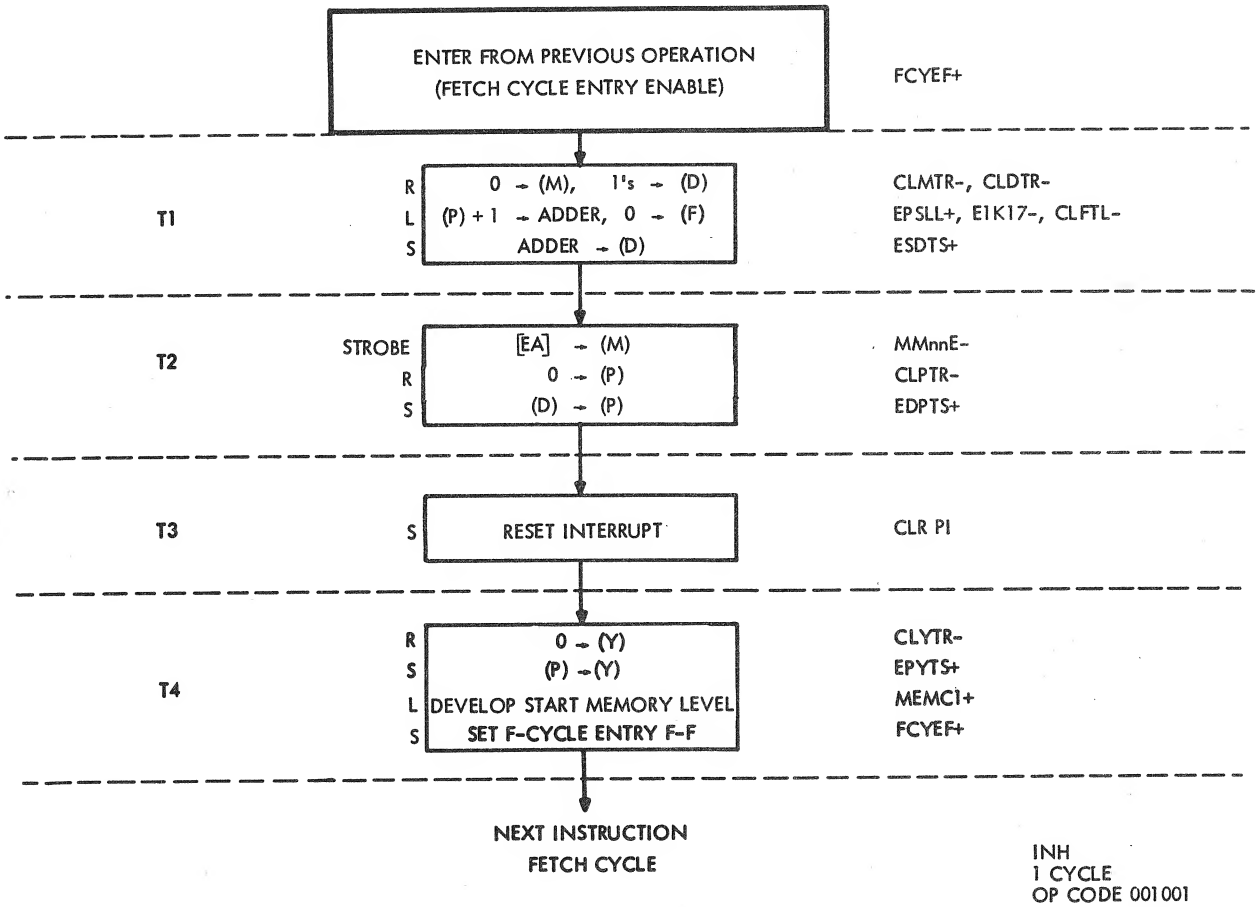
Description: $(A)_{9-16} \rightarrow (A)_{1-8}$
$0 \rightarrow (A)_{9-16}$




Execution Time ($\mu$sec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-J5 | 116-D7/D9 | Force carry to adder |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L10 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (EDPTL+)(MCRST+) * | 129-H10 | 101--116-H10 | Clear P-register |
| EDPTS- | 129-L9 | F | TL2 | S | (EDPTL+)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register into P-register |
| EASTL+ | 127-L1 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+)(M09FF+) | 127-G11 | 101--116-A5 | Enable A-register to adder |
| EMSHL+ | 127-L8 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+)(M09FF+) | 127-G11 | 101--108-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+)(M09FF+) | 127-G11 | 101--108-A9 | Enable M-(1-7) to adder |
| EMSLL+ | 127-L10 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+)(M09FF+) | 127-G11 | 109--116-A8 | Enable M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+)(M09FF+) | 127-G11 | 109--116-A9 | Enable M-(8-16) to adder |
| CLDTR- | 127-J5 | F | TL3 | R | (TL3FF+)(ACYLF-) | 125-D6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLATR- | 122-H7 | F | TL4 | R | (GEN0A+)(M11FF+)(TL4FF+) | 122-D5 | 101--116-H5 | Clear A-register |
| ETAHS+ | 122-L5 | F | TL4 | S | (GEN0A+)(M09FF+)(M11FF+) | 122-D5/H5 | 101--108-D4 | Enable D(9-16) into A(1-8) |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-D3 | 101--116-L11 | Clear Y-register |
| EPYTS+ | 129-L4 | F | TL4 | S | (P1SEX-)(E01NS+)(0PGJS-) | 129-D4/H4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

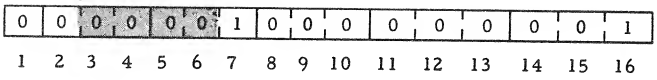
*See gate A1A44-F at 129-D8 for EDPTL+


ENTER FROM PREVIOUS OPERATION
(FETCH CYCLE ENTRY ENABLE)
FCYEF+
T1
R
L
S
0 → (M), 1's → (D)
(P) + 1 → ADDER, 0 → (F)
ADDER → (D)
CLMTR-, CLDTR-
EPSLL+, E1K17-, CLFTL-
ESDTS+
T2
STROBE
R
S
[EA] → (M)
0 → (P)
(D) → (P)
MMnnE-
CLPTR-
EDPTS+
T3
S
RESET INTERRUPT
CLR PI
T4
R
S
L
S
0 → (Y)
(P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY F-F
CLYTR-
EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
INH
1 CYCLE
OP CODE 001001
3810

Instruction: Inhibit Program Interrupt (INH)

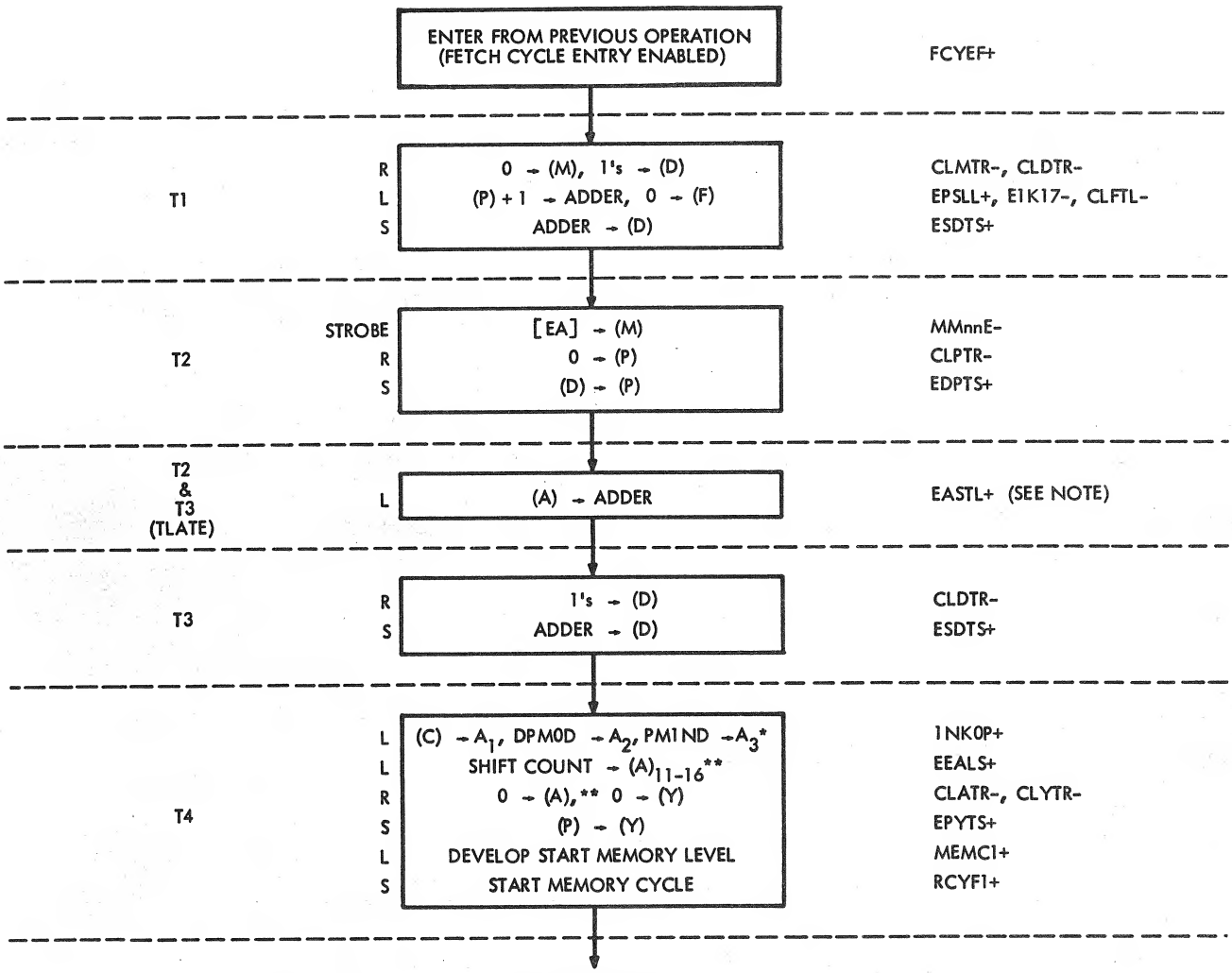
OP Code: 001001 Type: G, 1 cycle

Description: Set machine status to inhibit interrupt




Execution Time ($\mu$sec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-C3/ J6 | 116-D7-D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+) (MEMAC-)(MCRST+) | 129-D7/ H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+) (MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| CLR P1 | 134-A2 | F | TL3 | S | (GEN0B+)(M07FF+)(TL3FF+) (MCSET+) | 134-A2 | 134-E4 | Inhibit interrupt |
| CLYTR- | 129-J3 | F | TL4 | R | (SCZR0-)(TL4FF+)(MCRST+) | 129-D2 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+) (0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/ H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |




NOTE: MISSING SIGNALS CAN BE FOUND IN INK ANALYSIS

* APPLICABLE ONLY WHEN COMPUTER IS EQUIPPED WITH MEMORY EXPANSION OPTION

** APPLICABLE ONLY WHEN COMPUTER IS EQUIPPED WITH HIGH SPEED ARITHMETIC UNIT OPTION

3679

INK
1 CYCLE
OP CODE 000043

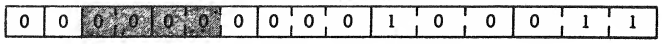
Instruction: Input Keys (INK)

OP Code: 000043 Type: G, 1 cycle

Description: (C) → $A_1$ (PM1) → $A_3$

(DP Mode) → $A_2$ 0 → $(A)_{4-11}$

Shift count → $(A)_{12-16}$

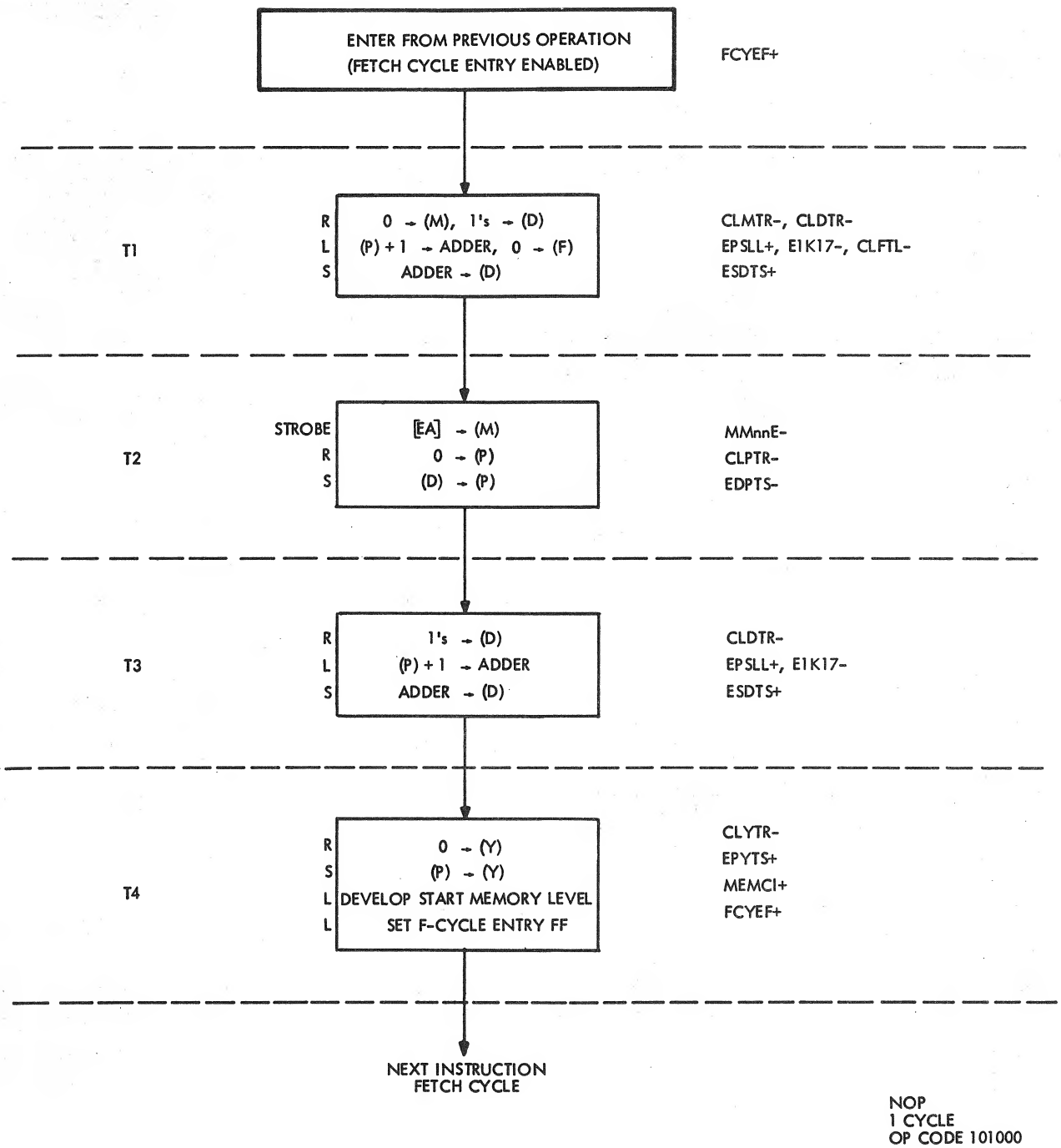



Execution Time (μsec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | TLATE- | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | TLATE- | L | (TLATE-) | 127-J5 | 116-D7/D9 | Force carry to adder |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L10 | Reset F-register<br>Reset shift counter<br>Reset AZZZZ FF |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Reset M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+)(MCRST+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+)(MCSET+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCRST+) | 129-D8 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register into P-register |
| EASTL+ | 127-L1 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G9 | 101--116-A5 | Enable A-register to adder |
| EMSHL+ | 127-L8 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G9 | 101--107-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G9 | 101--107-A9 | Enable M-(1-7) to adder |
| EMSLL+ | 127-L10 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G9 | 107--116-A8 | Enable M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G9 | 108--116-A9 | Enable M-(8-16) to adder |
| CLDTR- | 125-J6 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D6 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(ACYLF-)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| INK0P+ | 122-L8 | F | TL4 | L | (GEN0B+)(M11FF+)(M15FF+)(TL4FF+) | 122-G8 | 101--103-J6 | CB1TF into $A_1$<br>DPM0D into $A_2$<br>PM1ND into $A_3$* |
| EEALS+** | 122-K4 | A | TL4 | L | (GEN0B+)(M11FF+)(TL4FF+) | 122-D4 | 111--116-G4 | Enable E(11-16) into A(11-16) |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| CLATR-** | 122-H7 | A | TL4 | R | (GEN0B+)(M11FF+)(TL4FF+)(MCRST+) | 122-D4 | 101--116-H5 | Clear A-register |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 128-D3 | 101--116-L11 | Clear Y-register |
| EPYTS+ | 129-L4 | F | TL4 | S | (P1SEX-)(E01NS+)(0PGJS-)(MCSET+) | 129-D4/H4 | 101--116-K11 | Enable P-register to Y-register |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

*Applicable only when computer is equipped with Memory Expansion Option

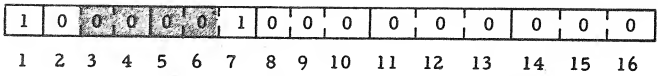
**Applicable only when computer is equipped with high speed Arithmetic Unit Option


ENTER FROM PREVIOUS OPERATION
(FETCH CYCLE ENTRY ENABLED)
FCYEF+
T1
R
L
S
0 → (M), 1's → (D)
(P) + 1 → ADDER, 0 → (F)
ADDER → (D)
CLMTR-, CLDTR-
EPSLL+, E1K17-, CLFTL-
ESDTS+
T2
STROBE
R
S
[EA] → (M)
0 → (P)
(D) → (P)
MMnnE-
CLPTR-
EDPTS-
T3
R
L
S
1's → (D)
(P) + 1 → ADDER
ADDER → (D)
CLDTR-
EPSLL+, E1K17-
ESDTS+
T4
R
S
L
L
0 → (Y)
(P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY FF
CLYTR-
EPYTS+
MEMCI+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
NOP
1 CYCLE
OP CODE 101000
5602

Instruction: No Operation (NOP)

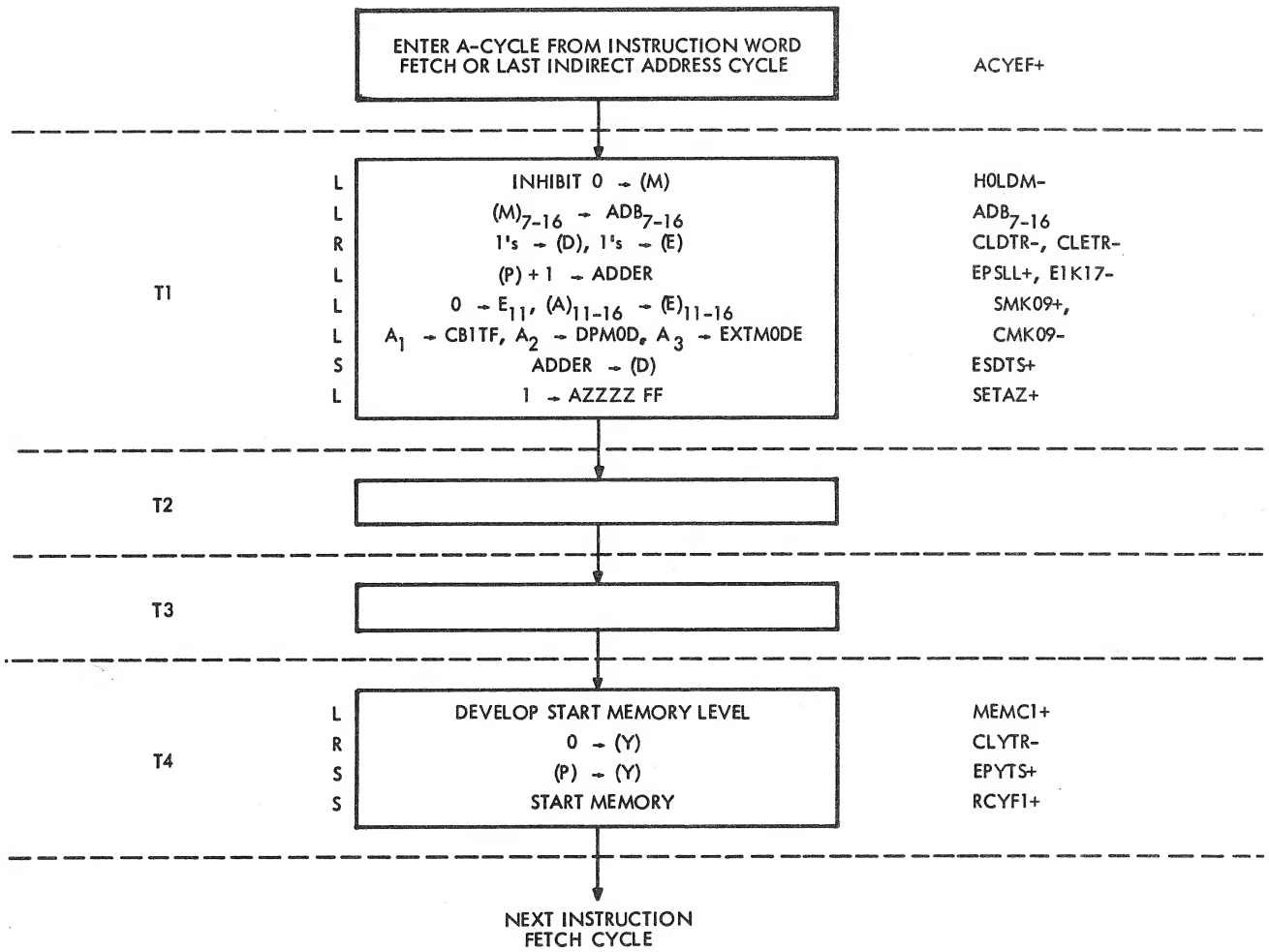
OP Code: 101000 Type: G, 1 cycle

Description: No operation




Execution Time ($\mu$sec): 0.96

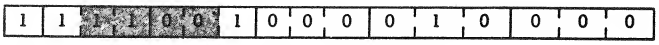
| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-H8 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-L4 | 116-D7-D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCRST+) | 129-D7/H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| EPSLL+ | 128-H4 | | | | (SKGRP+) | 128-H8 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | | | | (SKGRP+) | 127-J6 | 116-D7-D9 | Force carry to adder |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (10GRP-)(TL3FF+)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYLF-)(TL4FF+)(MCRST+) | 129-A1/D3 | 101--116-L11 | Clear P-register |
| EPYTS+ | 129-L4 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |


ENTER A-CYCLE FROM INSTRUCTION WORD FETCH OR LAST INDIRECT ADDRESS CYCLE
ACYEF+
T1
L INHIBIT 0 → (M) HOLDM-
L (M)7-16 → ADB7-16 ADB7-16
R 1's → (D), 1's → (E) CLDTR-, CLETR-
L (P) + 1 → ADDER EPSLL+, E1K17-
L 0 → E11, (A)11-16 → (E)11-16 SMK09+,
L A1 → CB1TF, A2 → DPM0D, A3 → EXTM0DE CMK09-
S ADDER → (D) ESDTS+
L 1 → AZZZZ FF SETAZ+
T2
T3
T4
L DEVELOP START MEMORY LEVEL MEMC1+
R 0 → (Y) CLYTR-
S (P) → (Y) EPYTS+
S START MEMORY RCYF1+
NEXT INSTRUCTION FETCH CYCLE
3678
OTK
2 CYCLES
OP CODE 171020

Instruction: Output Keys (OTK)

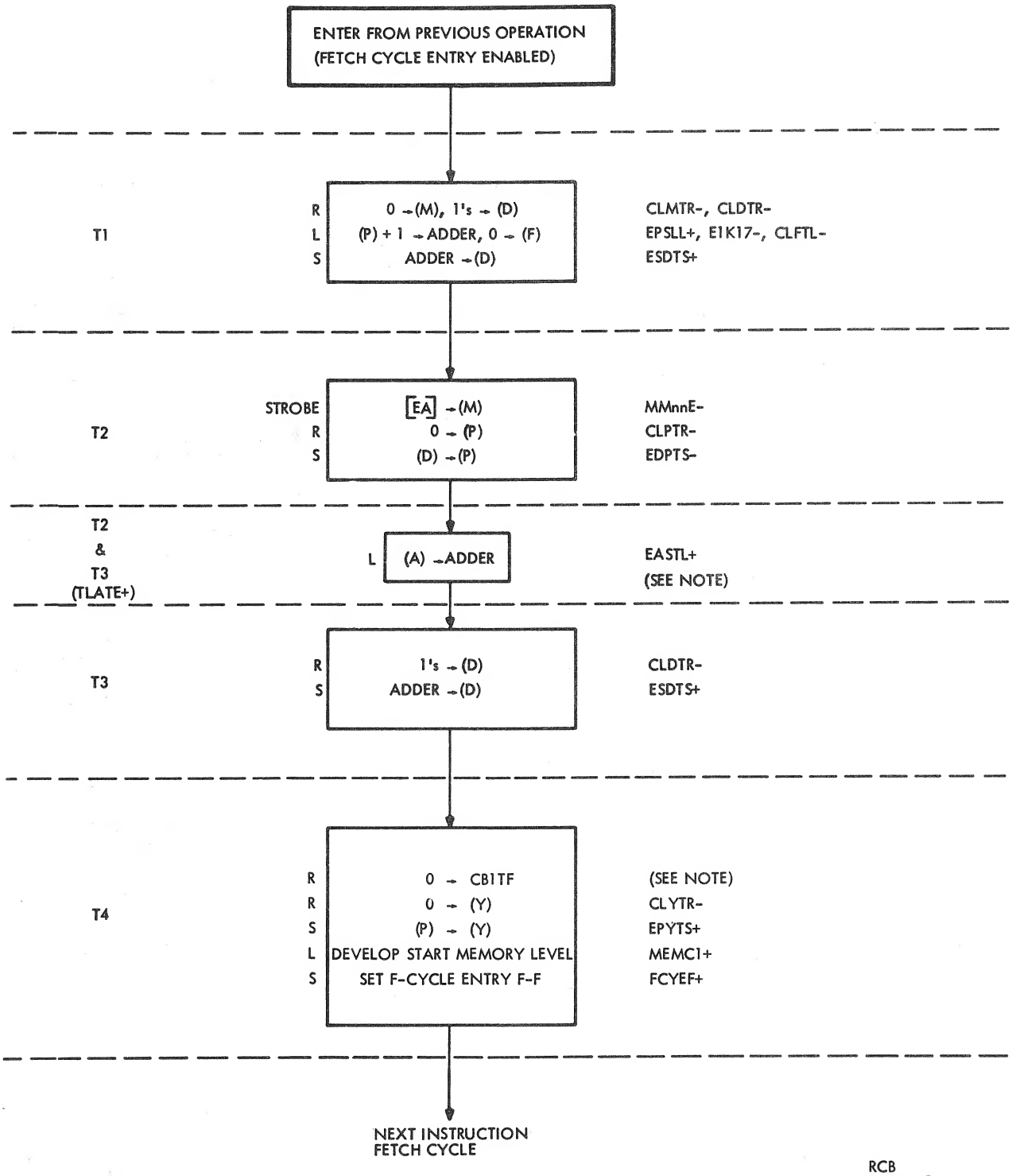
OP Code: 171020 Type: I/O, 2 cycles

Description: $A_1 \rightarrow (C)$
$A_2 \rightarrow$ (DP Mode)
$A_3 \rightarrow$ (Ext mode)
$0 \rightarrow E_{11}$
$(A)_{11-16} =$ Shift count $\rightarrow (E)_{11-16}$




Execution Time ($\mu$sec): 1.92

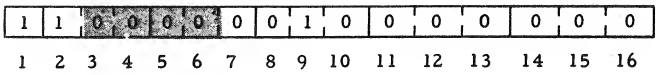
| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| ADB7-16 | 138-11 | | | | (DMCCY-)(M__FF+) | 138-11 | (See main frame wire list) | |
| 0TB01-16 | 138-E/L | | | | (A01FF+) through (A16FF+) | 138-C/J | (See main frame wire list) | |
| EPSLL+ | 128-H4 | | | | (10GRP+) | 128-D5 | 101--116-A10 | Enable P-register to adder |
| SMK09+ | 122-L9 | | | | (SMKXX+)(FCX00+)(ADB07+) | 122-G9 | 122-L7<br>111-L4<br>112-116-L4<br>124-F2 | Enable generate CMK09-<br>Reset E-register bit 11<br>Enable A(11-16) into E(11-16)<br>Enable reset CB1TF |
| H0LDM- | 128-H8 | | | | (10GRP+)(ACYEF+) | 128-H8 | 128-K8 | Inhibit M-register clear |
| CLDTR- | 125-J5 | A | TL1 | R | (ACYEF+)(TL1FF+)(JST0P-)(1RS0P-)(1MA0P-)(MCRST+) | 125-D4 | 101--116-E7 | Clear D-register to ONEs |
| E1K17- | 127-L4 | A | TL1 | L | (ACYLF+)(SUB0P-) | 127-A6/E7 | 116-D7-D9<br>117-B1 | Force carry to adder |
| CMK09- | 122-L7 | A | TL1 | L | (SMK09+)(TL1FF+) | 122-L7 | 124-F2<br>125-E2 | Enable set CB1TF<br>Enable generate CLETR- |
| ESDTS+ | 125-L4 | A | TL1 | S | (ACYEF+)(TL1FF+)(JST0P-)(1RS0P-)(1MA0P-)(MCSET+) | 125-D4 | 101--116-D4-D8 | Enable adder sum to D-register |
| SETAZ+ | 125-J9 | A | TL1 | L | (SMKXX-) | 125-J9 | 125-L9 | Set AZZZZ on trailing edge of MCSET |
| MEMC1+ | 126-J11 | A | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| CLYTR- | 129-J3 | A | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-B1/D1 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | A | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-)(MCSET+)(DMCRQ-) | 129-D4/H4 | 101--116-K11 | Enable P-register into Y-register |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |


ENTER FROM PREVIOUS OPERATION
(FETCH CYCLE ENTRY ENABLED)
T1
R
L
S
0 → (M), 1's → (D)
(P) + 1 → ADDER, 0 → (F)
ADDER → (D)
CLMTR-, CLDTR-
EPSLL+, E1K17-, CLFTL-
ESDTS+
T2
STROBE
R
S
[EA] → (M)
0 → (P)
(D) → (P)
MMnnE-
CLPTR-
EDPTS-
T2
&
T3
(TLATE+)
L
(A) → ADDER
EASTL+
(SEE NOTE)
T3
R
S
1's → (D)
ADDER → (D)
CLDTR-
ESDTS+
T4
R
R
S
L
S
0 → CB1TF
0 → (Y)
(P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY F-F
(SEE NOTE)
CLYTR-
EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
RCB
1 CYCLE
OP CODE 140200
NOTE: MISSING SIGNALS CAN BE
FOUND IN RCB ANALYSIS
3482

Instruction: Reset C to ZERO (RCB)

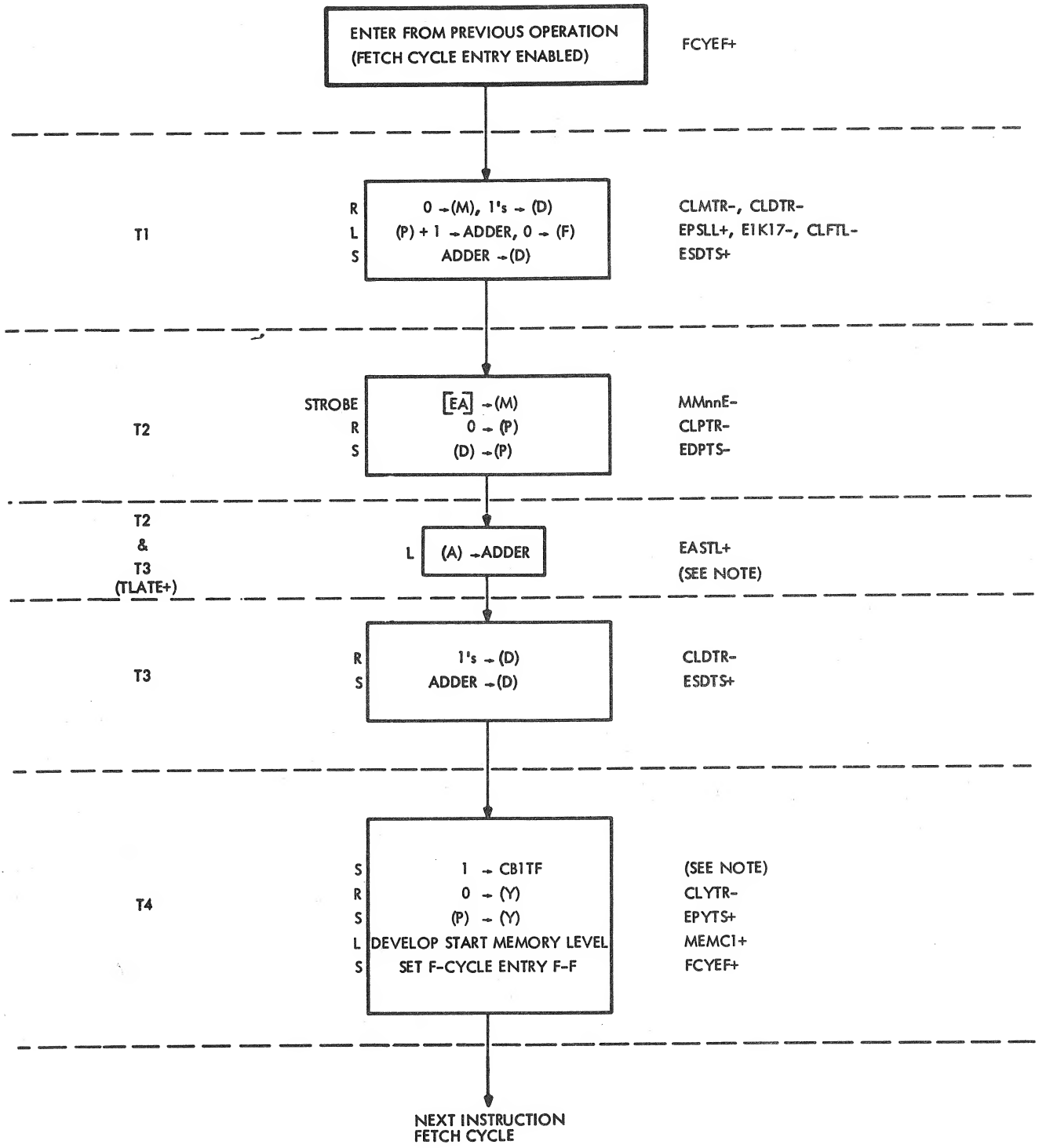
OP Code: 140200 Type: G, 1 cycle

Description: 0 → (C)




Execution Time (μsec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-C3/ J6 | 116-D7-D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+) (MEMAC-)(MCRST+) | 129-D7/ H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+) (MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| EASTL+ | 127-L1 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 101--116-A5 117 B/D | Enable A-register to adder |
| JAMKN- | 127-J3 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 101--116-D6 | Jam carry network |
| EMSHL+ | 127-L8 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 101--107-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 101--107-A9 | Enable M-(1-7) to adder |
| EMSLL+ | 127-L10 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 108--116-A8 | Enable M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 108--116-A9 | Enable M-(8-16) to adder |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| CB1TF | 124-L2 | F | TL4 | R | (GEN0A+)(M09FF+)(M11FF-) (TL4FF+)(D1V0P-)(MCRST+) | 124-A2 | 124-L1 | Reset CB1TF |
| CLYTR- | 129-J3 | F | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-D2 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+) (0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/ H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |




NOTE: MISSING SIGNALS CAN BE FOUND IN SCB ANALYSIS

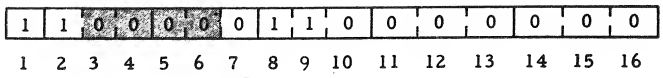
SCB
1 CYCLE
OP CODE 140600

3484

Instruction: Set C to ONE (SCB)

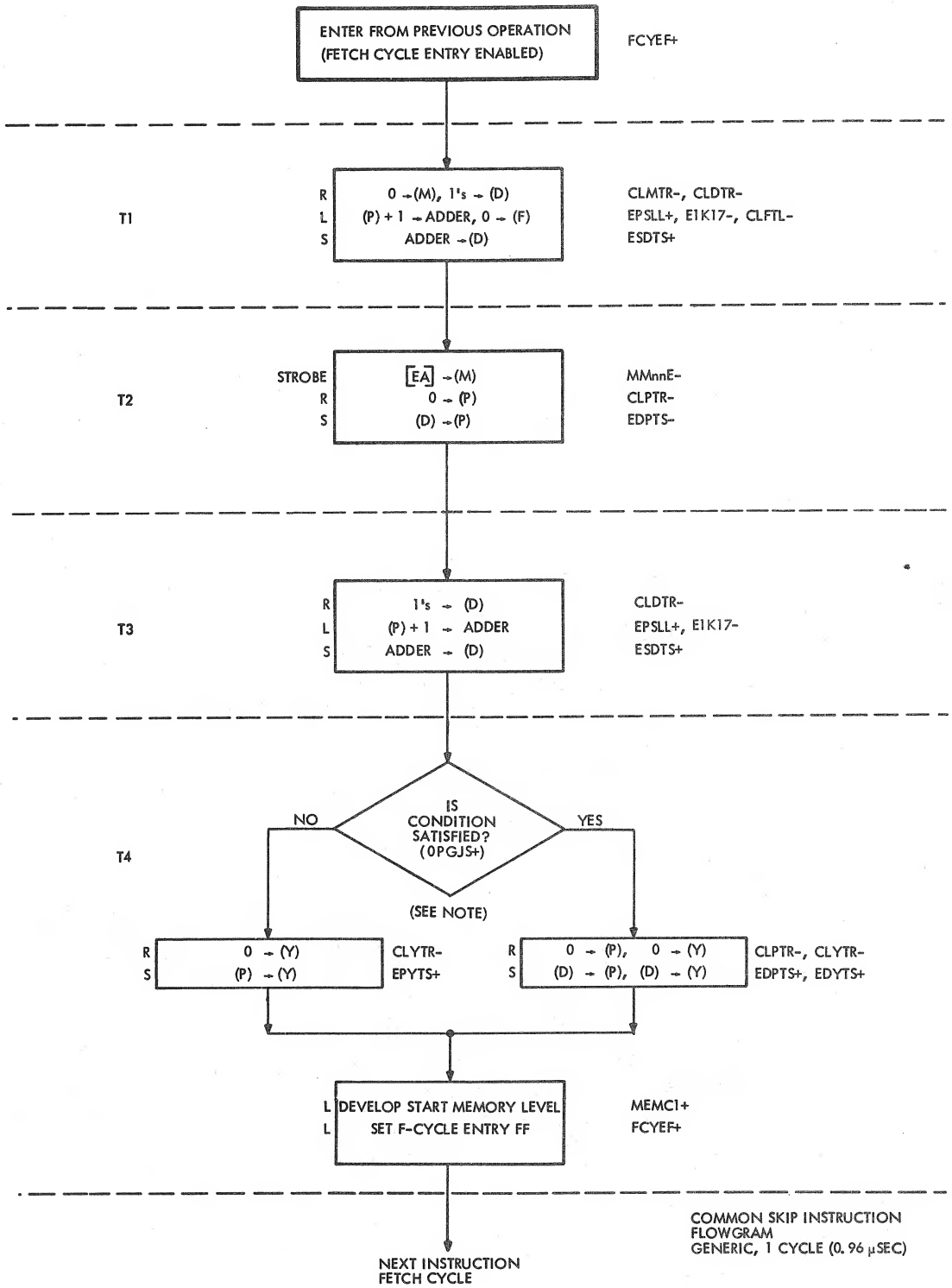
OP Code: 140600 Type: G, 1 cycle

Description: 1 → (C)




Execution Time (μsec): 0.96

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-C3/ J6 | 116-D7-D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+) (MEMAC-)(MCRST+) | 129-D7/ H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+) (MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| EASTL+ | 127-L1 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 101--116-A5 117 B/D | Enable A-register to adder |
| JAMKN- | 127-J3 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 101--116-D6 | Jam carry network |
| EMSHL+ | 127-L8 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 101--107-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 101--107-A9 | Enable M-(1-7) to adder |
| EMSLL+ | 127-L10 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 108--116-A8 | Enable M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+) (M09FF+) | 127-G11 | 108--116-A9 | Enable M-(8-16) to adder |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| CB1TF | 124-L2 | F | TL4 | S | (MCSET+)(GEN0A+)(TL4FF+) (M08FF+)(M09FF+) | 124-D6 | 124-L2 | Set CB1TF |
| CLYTR- | 129-J3 | F | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-D2 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+) (0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/ H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |




NOTE: REFER TO SKIP INSTRUCTION ANALYSES FOR SPECIFIC CONDITIONS

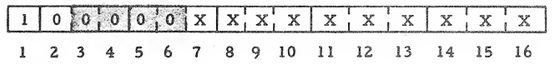
3488

Common Entry and Exit for Skip Instructions

Instruction:

OP Code: Type:

Description:

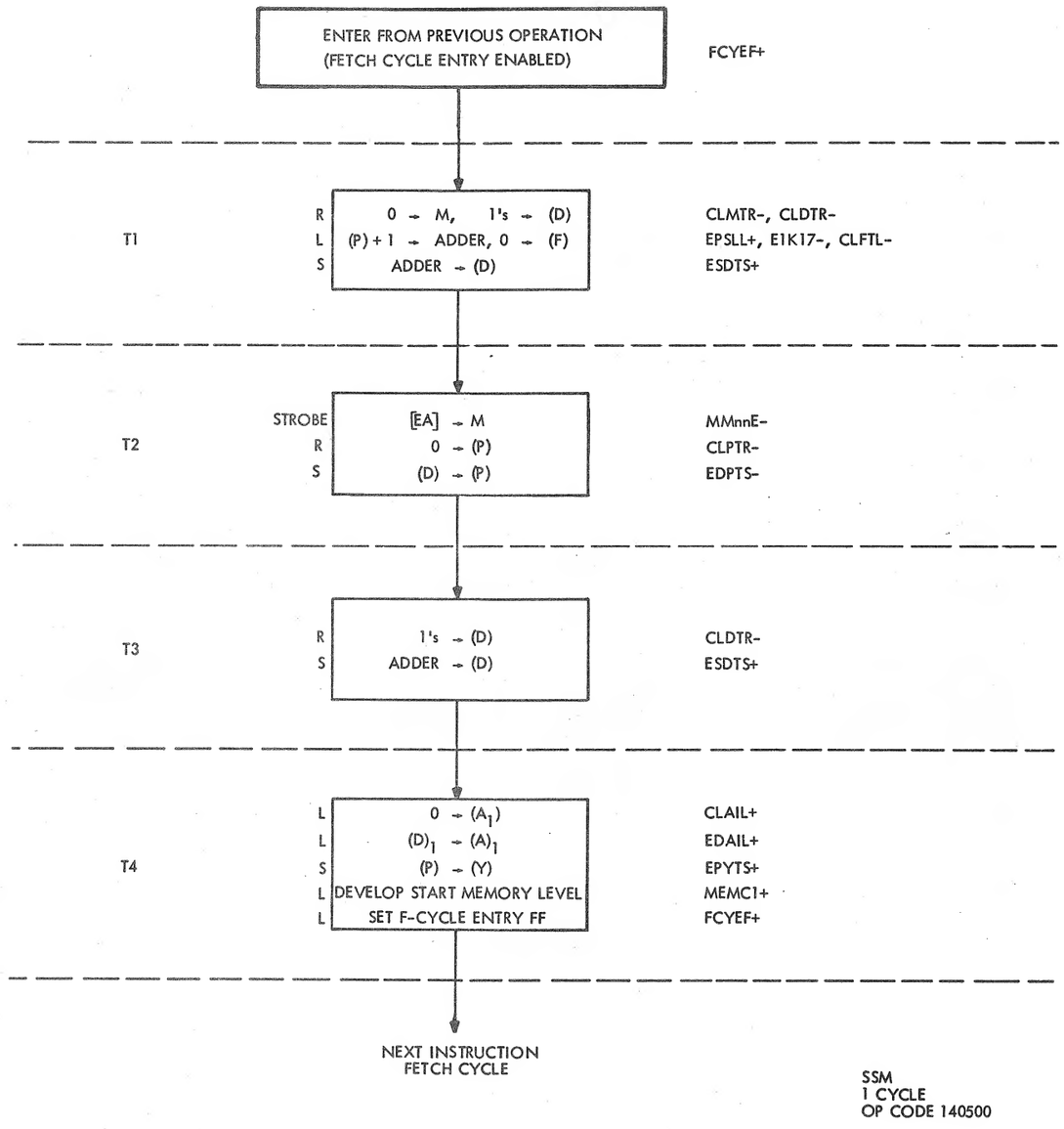



Execution Time ($\mu$sec):

| Function | Origin | Cyc | Tim | Clk | Boolean Expression | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | | | | (SKGRP-)V(TLATE-)(FCYEF+) | 128-H3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | | | | (SKGRP-)V(TLATE-) | 127-L4 | 116-D7-D9 | Force carry to adder |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ FF |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCRST+) | 129-D7/H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (10GRP-)(TL3FF+)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| 0PGJS+ | 129-J11 | | | | (See Table 2-1 for conditions) | | 129-D4-D9 | Condition satisfied? |
| CLPTR- | 129-J10 | F | TL4 | R | (0PGJS+)(E01NS+)(TL4FF+)(MCRST+) | 129-D7/H10 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL4 | S | (0PGJS+)(E01NS+)(TL4FF+)(MCSET+) | 129-D7/H9 | 101--116-H11 | Enable D-register into P-register |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYLF+)-(TL4FF+)(MCRST+) | 129-A1/D3 | 101--116-L11 | Clear Y-register |
| EDYTS+ | 129-L1 | F | TL4 | S | (ACYNX-)(MCSET+)(TL4FF+)(BREQ-)(0PGJS-) | 129-H1 | 101--116-G10 | Enable D-register into Y-register |
| EPYTS+ | 129-L4 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

Table 2-1.
Generating 0PGJS+ For Specific Skip Instructions

| Instruction | Op Code | Conditions for SKGRP+ | Origin |
|---|---|---|---|
| Skip If C Set (SSC) | 101001 | (CB1TF+)(M16FF+)(MEMAC-)(SKGRP+)(M07FF+) | 129-A9/F10 |
| Skip If C Reset (SRC) | 100001 | [(CB1TF+)(M16FF+)]-(MEMAC-)(SKGRP+)(M07FF-) | 129-A9/F11 |
| Skip If $A_{16} = 1$ (SLN) | 101100 | (A16FF+)(M10FF+)(MEMAC-)(SKGRP+)(M07FF+) | 129-A4/F10 |
| Skip If $A_{16} = 0$ (SLZ) | 100100 | [(A16FF+)(M10FF+)]-(MEMAC-)(M07FF-) | 129-A4/F11 |
| Skip If A Minus (SMI) | 101400 | (A01FF+)(M08FF+)(MEMAC-)(SKGRP+)(M07FF+) | 129-A11/F10 |
| Skip If A Plus (SPL) | 100400 | [(A01FF+)(M08FF+)]-(MEMAC-)(M07FF-) | 129-A11/F11 |
| Skip If A Not Zero (SNZ) | 101040 | (AZER0-)(M11FF+)(MEMAC-)(SKGRP+)(M07FF+) | 129-A10/F10 |
| Skip If A Zero (SZE) | 100040 | [(AZER0-)(M11FF+)]-(MEMAC-)(M07FF-) | 129-A10/F11 |
| Skip If Sense Switch 1 is Set (SS1) | 101020 | (SENS1+)(M12FF+)(MEMAC-)(SKGRP+)(M07FF+) | 129-A5/F10 |
| Skip If Sense Switch 1 is Reset (SR1) | 100020 | [(SENS1+)(M12FF+)]-(MEMAC-)(M07FF-) | 129-A5/F11 |
| Skip If Sense Switch 2 is Set (SS2) | 101010 | (SENS2+)(M13FF+)(MEMAC-)(SKGRP+)(M07FF+) | 129-A6/F10 |
| Skip If Sense Switch 2 is Reset (SR2) | 100010 | [(SENS2+)(M13FF+)]-(MEMAC-)(M07FF-) | 129-A6/F11 |
| Skip If Sense Switch 3 is Set (SS3) | 101004 | (SENS3+)(M14FF+)(MEMAC-)(SKGRP+)(M07FF+) | 129-A7/F10 |
| Skip If Sense Switch 3 is Reset (SR3) | 100004 | [(SENS3+)(M14FF+)]-(MEMAC-)(M07FF-) | 129-A7/F11 |
| Skip If Sense Switch 4 is Set (SS4) | 101002 | (SENS4+)(M15FF+)(MEMAC-)(SKGRP+)(M07FF+) | 129-A8/F10 |
| Skip If Sense Switch 4 is Reset (SR4) | 100002 | [(SENS4+)(M15FF+)]-(MEMAC-)(M07FF-) | 129-A8/F11 |
| Skip If Any Sense Switch Set (SSS) | 101036 | (SENS1+)(M12FF+)V(SENS2+)(M13FF+)V(SENS3+)(M14FF+)V(SENS4+)(M15FF+)Λ(MEMAC-)(SKGRP+)(M07FF+) | 129-A6/A7/A8/F10 |
| Skip If Any Sense Switch Reset (SSR) | 100036 | [(SENS1+)(M12FF+) V(SENS2+)(M13FF+)V(SENS3+)-(M14FF+)V(SENS4+)(M15FF+)]Λ(MEMAC-)(SKGRP+)(M07FF-) | 129-A6/A7/A8/F11 |
| Unconditional Skip | 100000 | (M08FF-) through (M16FF-)Λ(MEMAC-)(SKGRP+)(M07FF-) | 129-A3 through A11/F11 |

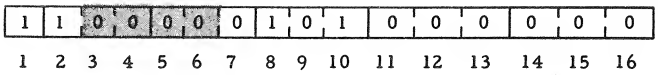

ENTER FROM PREVIOUS OPERATION
(FETCH CYCLE ENTRY ENABLED)
FCYEF+
T1
R
L
S
0 → M, 1's → (D)
(P) + 1 → ADDER, 0 → (F)
ADDER → (D)
CLMTR-, CLDTR-
EPSLL+, E1K17-, CLFTL-
ESDTS+
T2
STROBE
R
S
[EA] → M
0 → (P)
(D) → (P)
MMnnE-
CLPTR-
EDPTS-
T3
R
S
1's → (D)
ADDER → (D)
CLDTR-
ESDTS+
T4
L
L
S
L
L
0 → (A1)
(D)1 → (A)1
(P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY FF
CLAIL+
EDAIL+
EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
SSM
1 CYCLE
OP CODE 140500

3800

Instruction: Set Sign Minus (SSM)

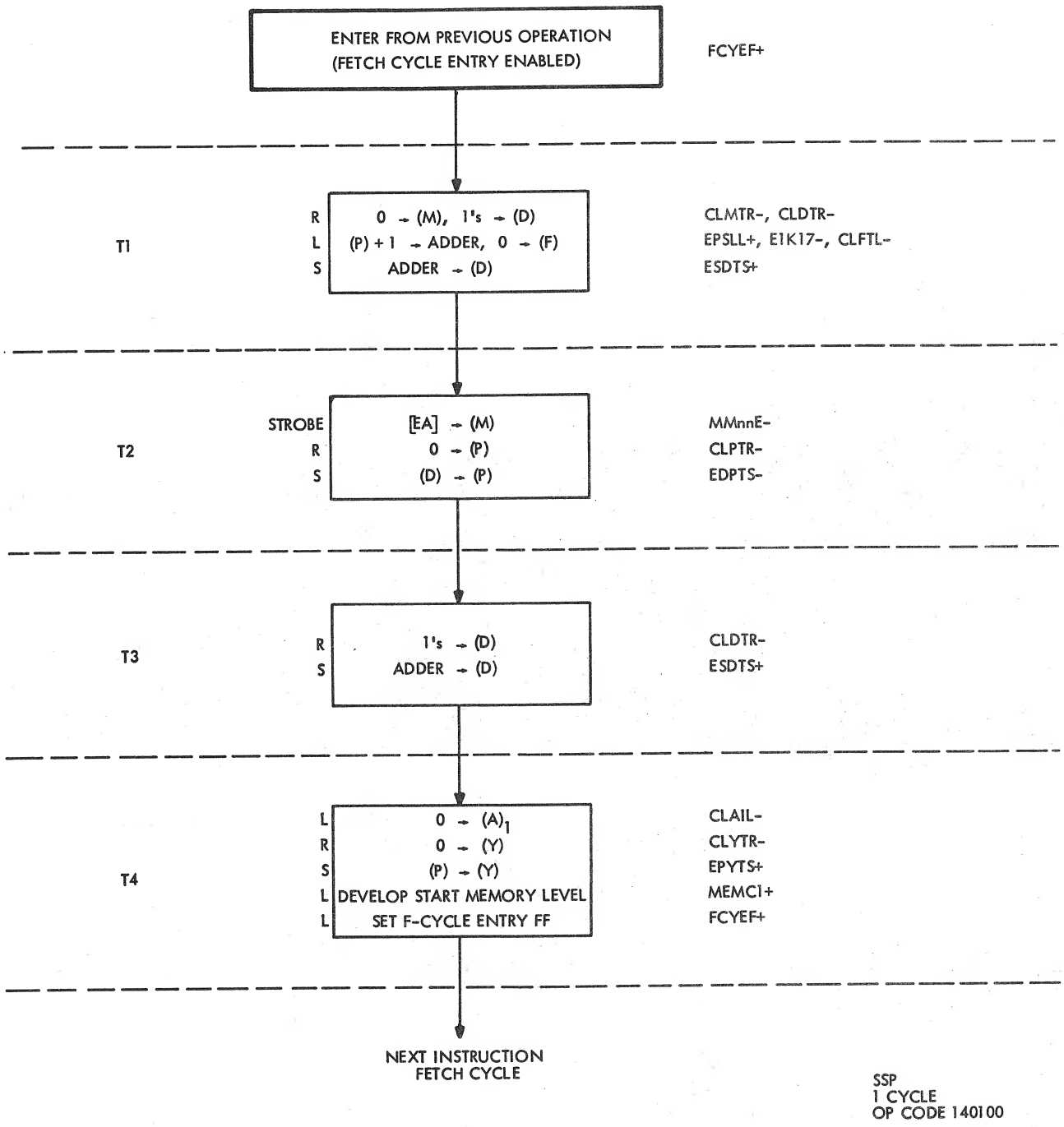
OP Code: 140500 Type: G, 1 cycle

Description: $1 \rightarrow (A)_1$




Execution Time ($\mu$sec): 0.96

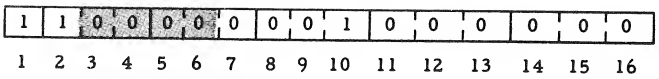
| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-C3/ J6 | 116-D7-D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+) (MEMAC-)(MCRST+) | 129-D7/ H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+) (MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| CLA1L+ | 130-H11 | F | TL4 | L | (GEN0A+)(TL4FF+)(M10FF+) | 130-F11 | 101-H3 | Clear A-register bit 1 |
| EDA1L+ | 130-H11 | F | TL4 | L | (GEN0A+)(TL4FF+)(M08FF+) (M10FF+) | 130-F10 | 101-H6 | Enable D-register bit 1 to A-register bit 1 |
| CLYTR- | 129-J3 | F | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-D2 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+) (0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/ H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |


ENTER FROM PREVIOUS OPERATION
(FETCH CYCLE ENTRY ENABLED)
FCYEF+
T1
R
L
S
0 → (M), 1's → (D)
(P) + 1 → ADDER, 0 → (F)
ADDER → (D)
CLMTR-, CLDTR-
EPSLL+, E1K17-, CLFTL-
ESDTS+
T2
STROBE
R
S
[EA] → (M)
0 → (P)
(D) → (P)
MMnnE-
CLPTR-
EDPTS-
T3
R
S
1's → (D)
ADDER → (D)
CLDTR-
ESDTS+
T4
L
R
S
L
L
0 → (A)1
0 → (Y)
(P) → (Y)
DEVELOP START MEMORY LEVEL
SET F-CYCLE ENTRY FF
CLAIL-
CLYTR-
EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
SSP
1 CYCLE
OP CODE 140100
3501

Instruction: Set Sign Plus (SSP)

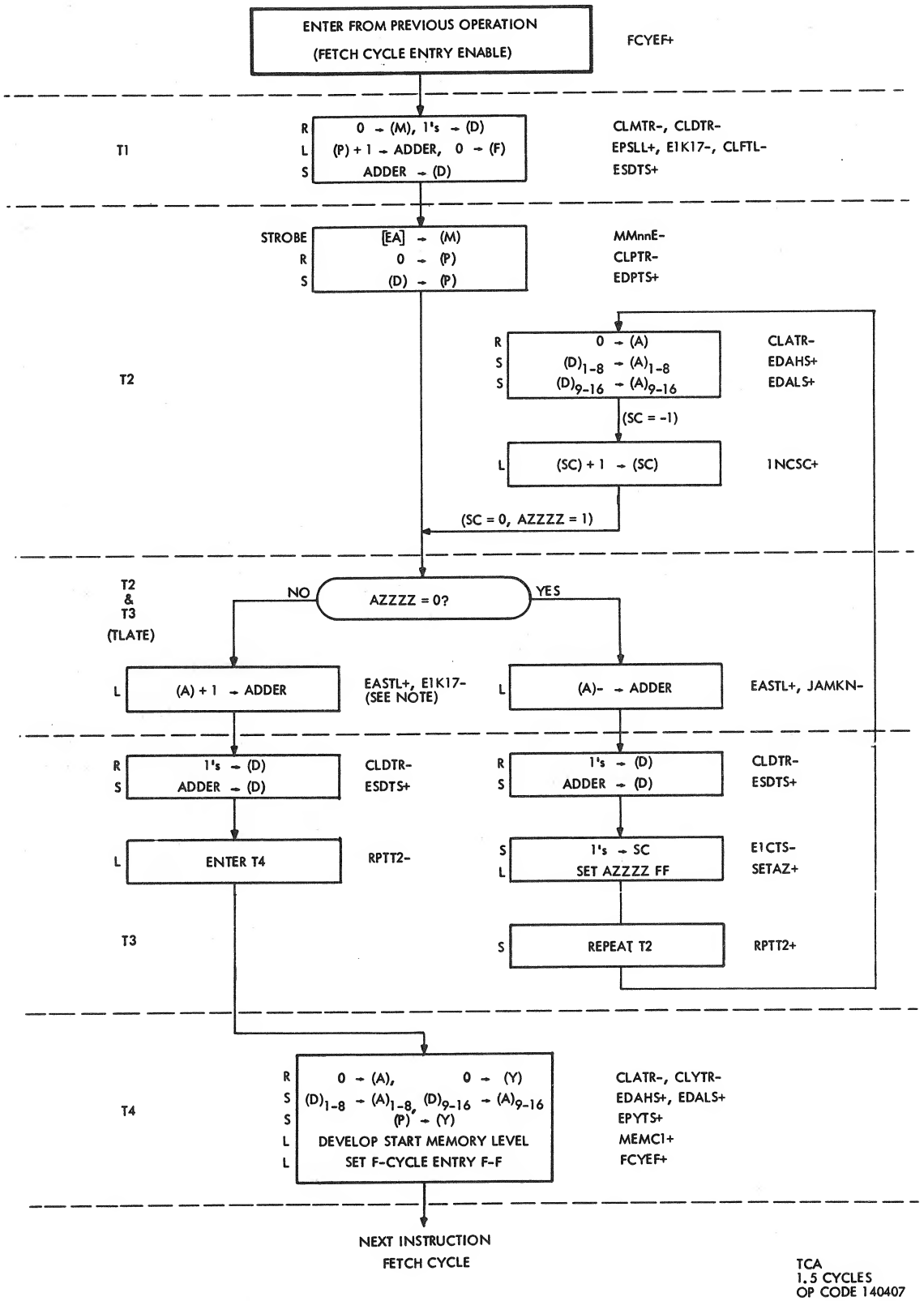
OP Code: 140100 Type: G, 1 cycle

Description: $0 \rightarrow (A)_1$




Execution Time ($\mu$sec): 0.96

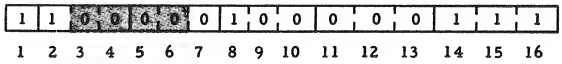
| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-C3/ J6 | 116-D7-D9 | Force carry to adder |
| CLMTR- | 129-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+) (MEMAC-)(MCRST+) | 129-D7/ H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+) (MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| CLA1L+ | 130-H11 | F | TL4 | L | (GEN0A+)(TL4FF+)(M10FF+) | 130-F11 | 101-H3 | Clear A-register bit 1 |
| CLYTR- | 129-J3 | F | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-D2 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+) (0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/ H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |




NOTE: MISSING SIGNALS CAN BE FOUND IN TCA ANALYSIS

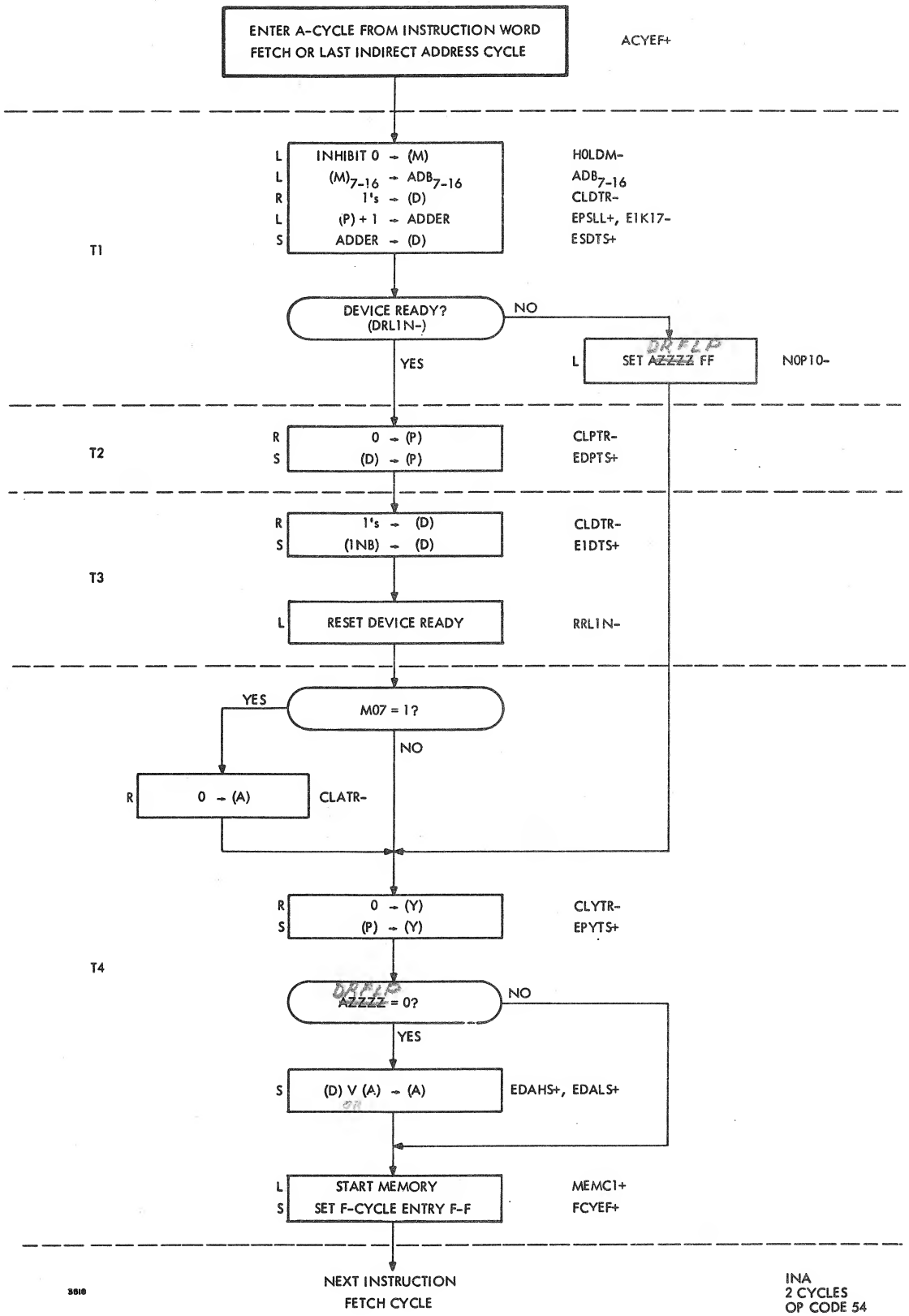
3609

Instruction: Two's Complement of A (TCA)
OP Code: 140407 Type: G, 1.5 cycles
Description: Two's Complement of (A) → (A)




Execution Time (μsec): 1.44

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | $\overline{\text{TLATE}}$ | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | $\overline{\text{TLATE}}$ | L | (TLATE-) | 127-C3/ J6 | 116-D7-D9 | Force carry to adder |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 120-B1<br>121-A5<br>125-L12 | Clear F-register<br>Clear shift counter<br>Clear AZZZZ F-F |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA-)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCRST+) | 129-D7/ H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (FCYEF+)(TL2FF+)(SCZR0+)(MEMAC-)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| EASTL+ | 127-L1 | F | TLATE | L | (AZZZZ-)(TLATE+)(GEN0P+)(M16FF+)(M02FF+)(M01FF+) | 127-C2 | 101--116-A5 | Enable A-register to adder |
| JAMKN- | 127-J3 | F | TLATE | L | (AZZZZ-)(TLATE+)(GEN0P+)(M16FF+)(M02FF+)(M01FF+) | 127-C2 | 101--116-D6<br>117 B/D | Jam carry network |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| SETAZ+ | 125-L10 | F | TL3 | L | (GEN0A+)(TL3FF+)(M08FF+)(M15FF+) | 125-A8 | 125-L10 | Set AZZZZ F-F |
| E1CTS- | 125-H8 | F | TL3 | S | (GEN0A+)(TL3FF+)(M08FF+)(M15FF+)(AZZZZ-)(MCSET+) | 125-A8/ H8 | 121-A7<br>126-F4 | Set shift counter to all ONEs generate RPTT2+ |
| RPTT2+ | 126-G4 | F | TL3 | S | E1CTS- | 126-G4 | 126-G4 | Repeat T2 |
| CLATR- | 122-H7 | F | TL2 | R | (GEN0A+)(TL2FF+)(AZZZZ+)(MCRST+) | 122-A5/ H7 | 101--116-H5 | Clear A-register |
| EDAHS+ | 122-L1 | F | TL2 | S | [(GEN0A+)(TL2FF+)(AZZZZ+) | 122-A5/ H1 | 101--108-G7 | Enable D(1-8) into A(1-8) |
| EDALS+ | 122-L2 | F | TL2 | S | (MCSET+)] | | 109--116-G7 | Enable D(9-16) into A(9-16) |
| INCSC+ | 126-L3 | F | TL2 | L | (FCYEF+)(TL2FF+) | 126-H3 | 121-A5 | Increment shift counter |
| EASTL+ | 127-L1 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+)(AZZZZ+) | 127-G11 | 101--116-A5 | Enable A-register to adder |
| EMSHL+ | 127-L8 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+)(AZZZZ+) | 127-G11 | 101--107-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+)(AZZZZ+) | 127-G11 | 101--107-A9 | Enable M-(1-7) to adder |
| EMSLL+ | 127-L10 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+)(AZZZZ+) | 127-G11 | 108--116-A8 | Enable M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE | L | (GEN0P+)(TLATE+)(M02FF+)(AZZZZ+) | 127-G11 | 108--116-A9 | Enable M-(8-16) to adder |
| E1K17- | 127-L4 | F | TLATE | L | (JAMKN-) | 127-J3 | 116-D7 | Force carry to adder |
| CLATR- | 122-H7 | F | TL4 | R | (GEN0A+)(M16FF+)(TL4FF+) | 122-A3 | 101--116-H5 | Clear A-register |
| EDAHS+ | 122-L1 | F | TL4 | S | (GEN0A+)(M16FF+)(TL4FF+) | 122-A3 | 101--108-G7 | Enable D(1-8) into A(1-8) |
| EDALS+ | 122-L2 | F | TL4 | S | (GEN0A+)(M16FF+)(TL4FF+) | 122-A3 | 109--116-G7 | Enable D(9-16) into A(9-16) |
| CLYTR- | 129-J3 | F | TL4 | R | (SCZR0+)(TL4FF+)(MCRST+) | 129-D2 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | F | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-)(MCSET+) | 129-D4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/ H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

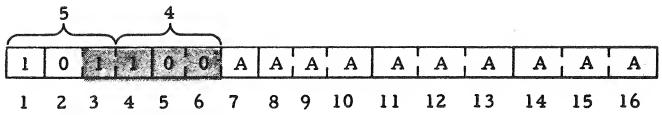

ENTER A-CYCLE FROM INSTRUCTION WORD FETCH OR LAST INDIRECT ADDRESS CYCLE
ACYEF+
T1
L INHIBIT 0 → (M) HOLDM-
L (M)7-16 → ADB7-16 ADB7-16
R 1's → (D) CLDTR-
L (P) + 1 → ADDER EPSLL+, E1K17-
S ADDER → (D) ESDTS+
DEVICE READY? (DRL1N-)
NO
YES
L SET AZZZZ FF N0P10-
T2
R 0 → (P) CLPTR-
S (D) → (P) EDPTS+
T3
R 1's → (D) CLDTR-
S (1NB) → (D) E1DTS+
L RESET DEVICE READY RRL1N-
M07 = 1?
YES
NO
R 0 → (A) CLATR-
T4
R 0 → (Y) CLYTR-
S (P) → (Y) EPYTS+
AZZZZ = 0?
NO
YES
S (D) V (A) → (A) EDAHS+, EDALS+
L START MEMORY MEMC1+
S SET F-CYCLE ENTRY F-F FCYEF+
NEXT INSTRUCTION FETCH CYCLE
INA
2 CYCLES
OP CODE 54

Instruction: Input to A (INA)

OP Code: 54 Type: 10, 2 cycles

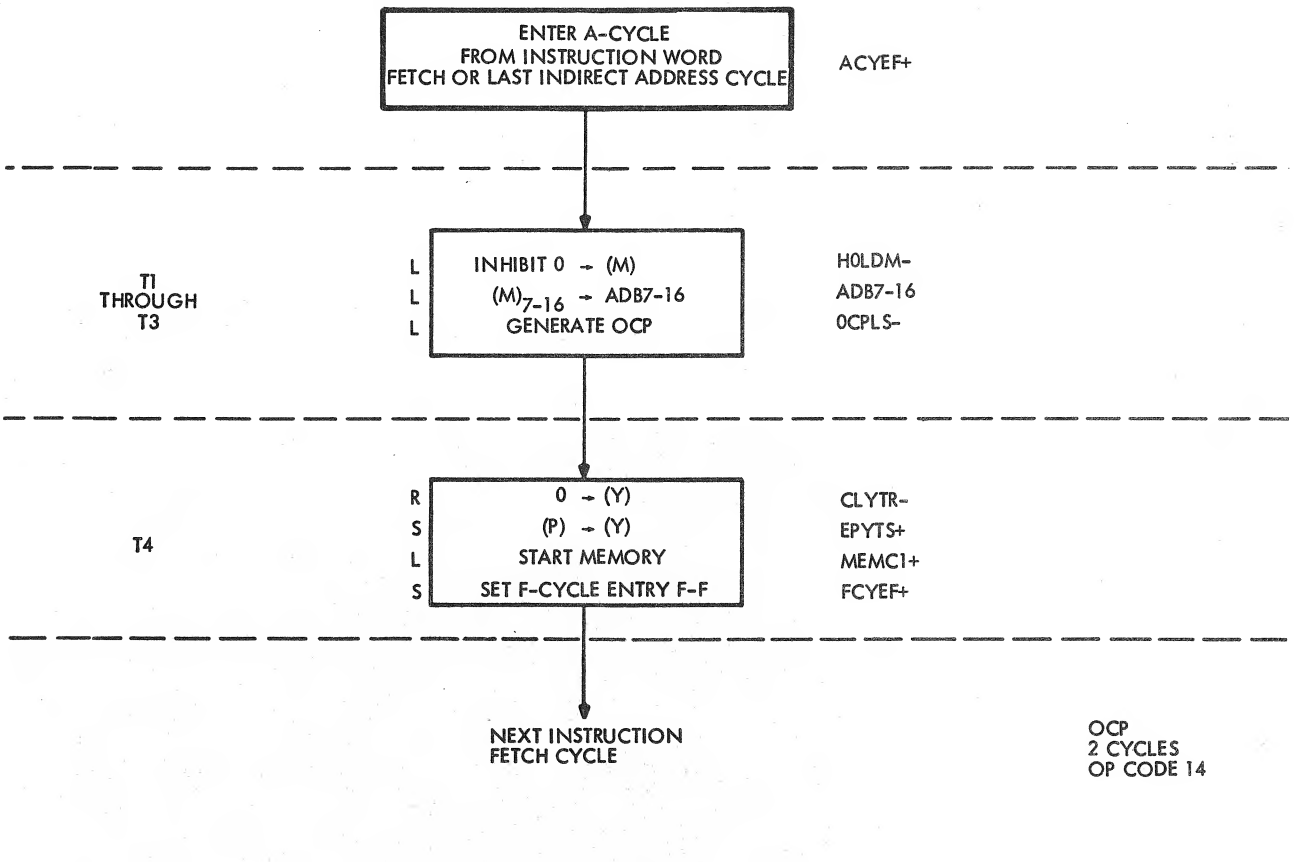
Description: (INB) → (A), $(1W)_7 = 1$

(INB) V (A) → (A), $(1W)_7 = 0$




Execution Time (μsec): 1.92

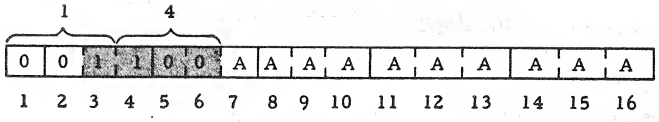
| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| MEMC1- | 126-J11 | F | TL4 | L | (10GRP+)(FCYLF+) | 126-D11 | 126-F11 | Block start memory |
| H0LDM- | 128-H8 | A | | | (10GRP+)(ACYEF+) | 128-H8 | 128-K8 | Inhibit M-register clear |
| ADB7-16 | 138-11 | A | | | (DMCCY-)(M__FF+) | 138-A11 | 143-XX | (See main frame wire list) |
| CLDTR- | 125-J5 | A | TL1 | R | (ACYEF+)(TL1FF+)(JST0P-)(1RS0P-)(1MA0P-)(MCRST+) | 125-D4 | 101--116-E7 | Clear D-register to ONEs |
| EPSLL+ | 128-H4 | A | TL1 | L | (10GRP+) | 128-D5 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | A | TL1 | L | (TLATE-) | 127-J6<br>127-J5 | 116-D7-D9<br>117-B1 | Force carry to adder |
| ESDTS+ | 125-L4 | A | TL1 | S | (ACYEF+)(TL1FF+)(JST0P-)(1RS0P-)(1MA0P-)(MCSET+) | 125-D4 | 101--116-D4-D8 | Enable adder sum to D-register |
| DRL1N- | 143-B3 | A | | | Function of IO option | | 125-D10 | Device ready line |
| N0P10- | 125-D10 | A | TL1 | L | (ACYEF+)(TL1FF+)(10GRP+)(DRL1N-) | 125-D10 | 125-L9 | Set AZZZZ F-F |
| CLPTR- | 129-J10 | A | TL2 | R | (TL2FF+)(10GRP+)(AZZZZ-)(0CPLS-)(MCRST+) | 129-F8 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | A | TL2 | S | (TL2FF+)(10GRP+)(AZZZZ-)(0CPLS-)(MCSET+) | 129-F8 | 101--116-H11 | Enable D-register to P-register |
| CLDTR- | 125-J5 | A | TL3 | R | (ANA0P-)(TL3FF+)(MCRST+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| E1DTS+ | 125-L6 | A | TL3 | S | (MCSET+)(RRL1N-) | 125-H6 | 101--116-D4 | Enable $1NB_{1-16}$ to D-register |
| RRL1N- | 134-K8 | A | TL24 | L | (M01FF+)(TL24F+)(AZZZZ-) | 134-G8 | 143-D5 | Reset ready to device |
| CLATR- | 122-H7 | A | TL4 | R | (ACYLF+)(TL4FF+)(10GRP+)(AZZZZ-)(M02FF-)(M07FF+)(M01FF+)(MCRST+) | 122-D7 | 101--116-H5 | Clear A-register if $1W_7 = 1$ |
| EDAHS+ | 122-L1 | A | TL4 | S | (ACYLF+)(TL4FF+)(10GRP+)(AZZZZ-)(M01FF+)(M02FF-)(MCSET+) | 122-D1 | 101--108-G7 | Enable D-register to A(1-8) |
| EDALS+ | 122-L2 | A | TL4 | S | (AZZZZ-)(M01FF+)(M02FF-)(MCSET+)(ACYLF+)(TL4FF+)(10GRP+) | 122-D1 | 109--116-G7 | Enable D-register to A(9-16) |
| CLYTR- | 129-J3 | A | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-B1/D2 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | A | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-)(MCSET+)(DMCRQ-) | 129-D4/H4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |


ENTER A-CYCLE
FROM INSTRUCTION WORD
FETCH OR LAST INDIRECT ADDRESS CYCLE
ACYEF+
T1
THROUGH
T3
L
L
L
INHIBIT 0 → (M)
(M)7-16 → ADB7-16
GENERATE OCP
HOLDM-
ADB7-16
OCPLS-
T4
R
S
L
S
0 → (Y)
(P) → (Y)
START MEMORY
SET F-CYCLE ENTRY F-F
CLYTR-
EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION
FETCH CYCLE
OCP
2 CYCLES
OP CODE 14
3496

Instruction: Output Control Pulse (OCP)

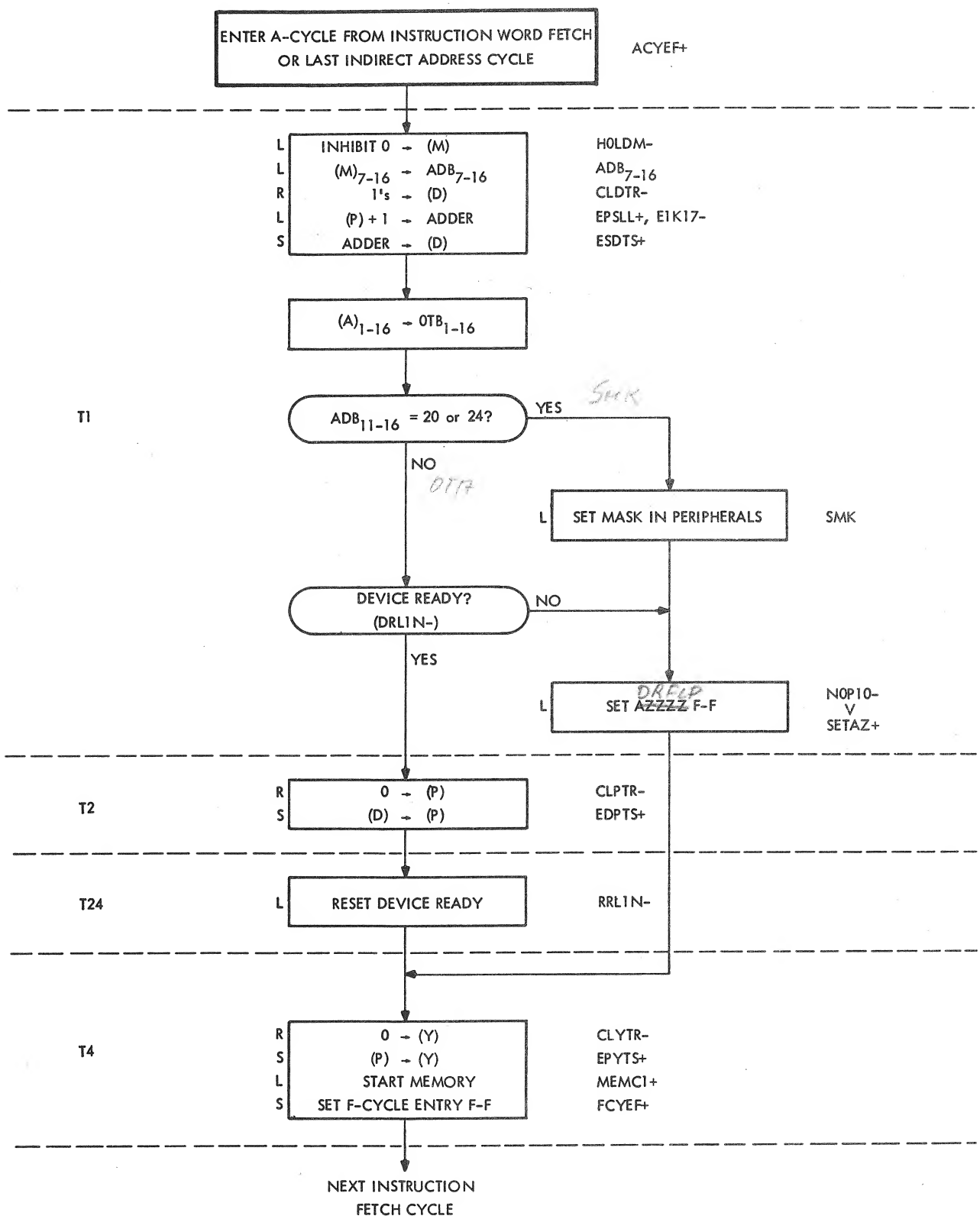
OP Code: 14 Type: 10, 2 cycles

Description: Set or reset function of specified device




Execution Time ($\mu$sec): 1.92

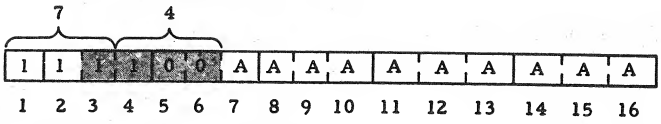
| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| MEMC1- | 126-J11 | F | TL4 | L | (10GRP+)(FCYLF+) | 126-D11 | 126-F11 | Block start memory |
| H0LDM- | 128-H8 | | | | (10GRP+)(ACYEF+) | 128-H8 | 128-K8 | Inhibit M-register clear |
| ADB7-16 | 138-11 | | | | (DMCCY-)(M__FF+) | 138-11 | See main frame wire list) | |
| 0CPLS- | 134-K10 | | | | (RESTR-)(FCYEF-)(10GRP+) (M01FF-)(M02FF-) | 134-C7 | 143-B4 | Control pulse to device |
| CLYTR- | 129-J3 | A | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-B1/ D1 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | A | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+) (0PGJS-)(MCSET+)(DMCRQ-) | 129-D4/ H4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/ H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |


ENTER A-CYCLE FROM INSTRUCTION WORD FETCH OR LAST INDIRECT ADDRESS CYCLE
ACYEF+
L INHIBIT 0 → (M)
L (M)7-16 → ADB7-16
R 1's → (D)
L (P) + 1 → ADDER
S ADDER → (D)
HOLDM-
ADB7-16
CLDTR-
EPSLL+, E1K17-
ESDTS+
(A)1-16 → OTB1-16
T1
ADB11-16 = 20 or 24?
YES
SMK
NO
OTA
L SET MASK IN PERIPHERALS
SMK
DEVICE READY? (DRL1N-)
NO
YES
L SET AZZZZ F-F
DRELP
NOP10-
V
SETAZ+
T2
R 0 → (P)
S (D) → (P)
CLPTR-
EDPTS+
T24
L RESET DEVICE READY
RRL1N-
T4
R 0 → (Y)
S (P) → (Y)
L START MEMORY
S SET F-CYCLE ENTRY F-F
CLYTR-
EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION FETCH CYCLE
3816
OTA/SMK
2 CYCLES
OP CODE 74

Instruction: Output from A (OTA)

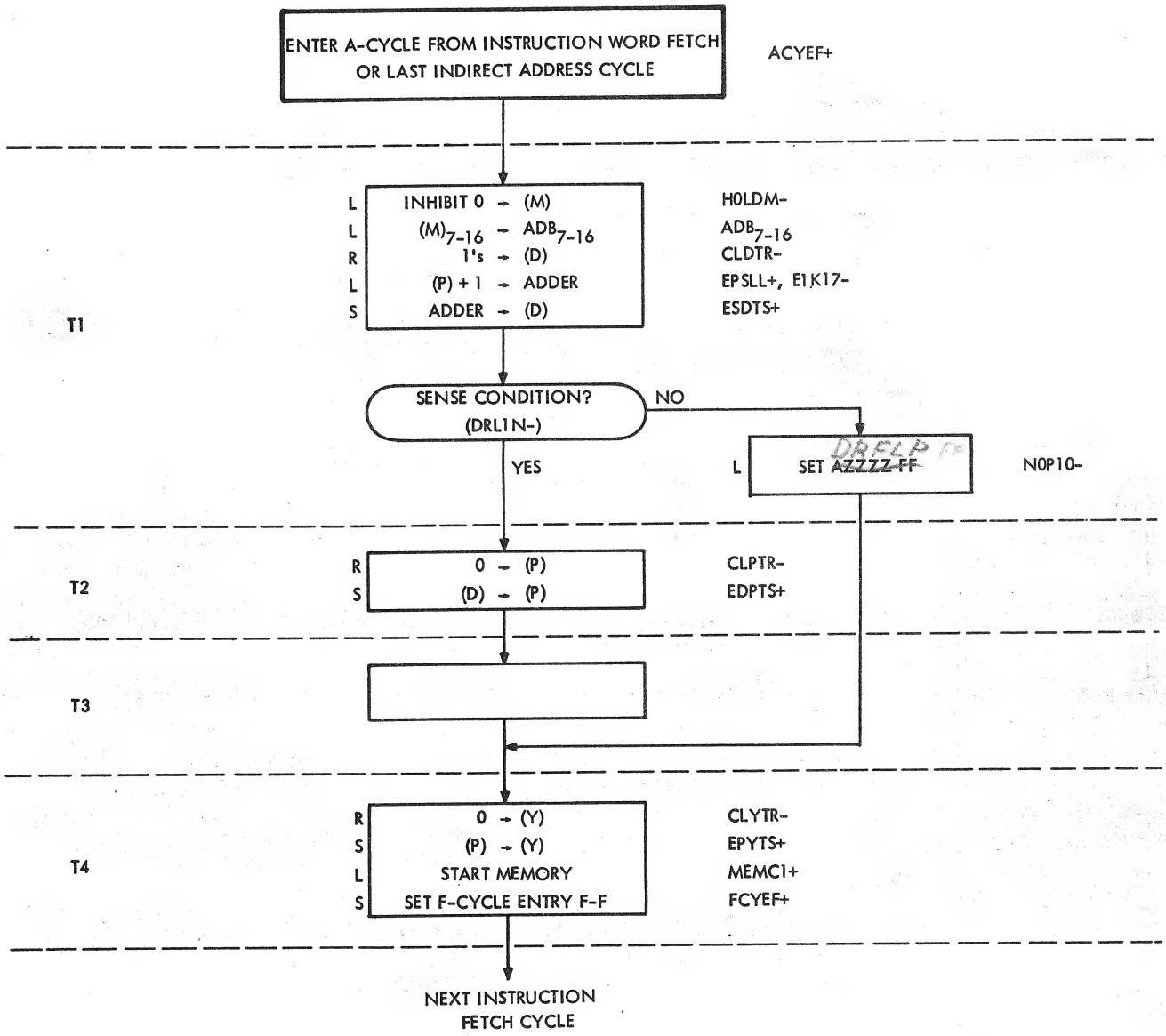
OP Code: 74 Type: 10, 2 cycles

Description: If ready: (A) → (0TB) and skip
If not ready: No output and no skip




Execution Time (μsec): 1.92

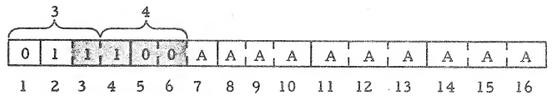
| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| MEMC1- | 126-J11 | F | TL4 | L | (10GRP+)(FCYLF+) | 126-D11 | 126-F11 | Block start memory |
| H0LDM- | 128-H8 | A | | | (10GRP+)(ACYEF+) | 128-H8 | 128-K8 | Inhibit M-register clear |
| EPSLL+ | 128-H4 | | | | (10GRP+) | 128-D5 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | A | TL1 | L | (TLATE-) | 127-J6 | 116-D7-D9 117-B1 | Force carry to adder |
| ADB7-16 | 138-11 | A | | | (DMCCY-)(M__FF+) | 138-11 | (See main frame wire list) | |
| CLDTR- | 125-J5 | A | TL1 | R | (ACYEF+)(TL1FF+)(JST0P-) (1RS0P-)(1MA0P-)(MCRST+) | 125-D4 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | A | TL1 | S | (ACYEF+)(TL1FF+)(JST0P-) (1RS0P-)(1MA0P-)(MCSET+) | 125-D4 | 101--116-D4-D8 | Enable adder sum to D-register |
| 0TB01-16 | 138-E/L | A | | | (A01FF+) through (A16FF+) | 138-C/J | (See main frame wire list) | |
| SMK01+ | 134-L11 | A | | | (SMKXX+)(ADB07-)(FCX00+) | 134-K11 | 143-B5 | ADB 7-16 = $0020_8$ (Set mask) |
| DRL1N- | 143-B3 | A | | | Function of I/O option | | 125-D10 | Device ready line |
| N0P10- | 125-D10 | A | TL1 | L | (ACYEF+)(TL1FF+)(10GRP+) (DRL1N-) | 125-D10 | 125-L9 | Set AZZZZ FF |
| | | | | | or | | | |
| SETAZ+ | 125-J10 | A | | | (SMKXX-) | 125-H9 | 125-L9 | Set AZZZZ FF |
| CLPTR- | 129-J10 | A | TL2 | R | (TL2FF+)(10GRP+)(AZZZZ-) (0CPLS-)(MCRST+) | 129-F8 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | A | TL2 | S | (TL2FF+)(10GRP+)(AZZZZ-) (0CPLS-)(MCSET+) | 129-F8 | 101--116-H11 | Enable D-register to P-register |
| RRL1N- | 134-K8 | A | TL24 | L | (M01FF+)(TL24F+)(AZZZZ-) | 134-G8 | 143-D5 | Reset ready to device |
| CLYTR- | 129-J3 | A | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-B1/ D1 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | A | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+) (0PGJS-)(MCSET+)(DMCRQ-) | 129-D4/ H4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/ H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |


ENTER A-CYCLE FROM INSTRUCTION WORD FETCH OR LAST INDIRECT ADDRESS CYCLE
ACYEF+
T1
L INHIBIT 0 → (M)
L (M)7-16 → ADB7-16
R 1's → (D)
L (P) + 1 → ADDER
S ADDER → (D)
HOLDM-
ADB7-16
CLDTR-
EPSLL+, E1K17-
ESDTS+
SENSE CONDITION? (DRL1N-)
NO
YES
L SET AZZZZ-FF
N0P10-
T2
R 0 → (P)
S (D) → (P)
CLPTR-
EDPTS+
T3
T4
R 0 → (Y)
S (P) → (Y)
L START MEMORY
S SET F-CYCLE ENTRY F-F
CLYTR-
EPYTS+
MEMC1+
FCYEF+
NEXT INSTRUCTION FETCH CYCLE
SKS
2 CYCLES
OP CODE 34
3817

Instruction: Skip if Ready Line Set (SKS)

OP Code: 34 Type: 10, 2 cycles

Description: If f(device) is satisfied, device is ready and next instruction is skipped



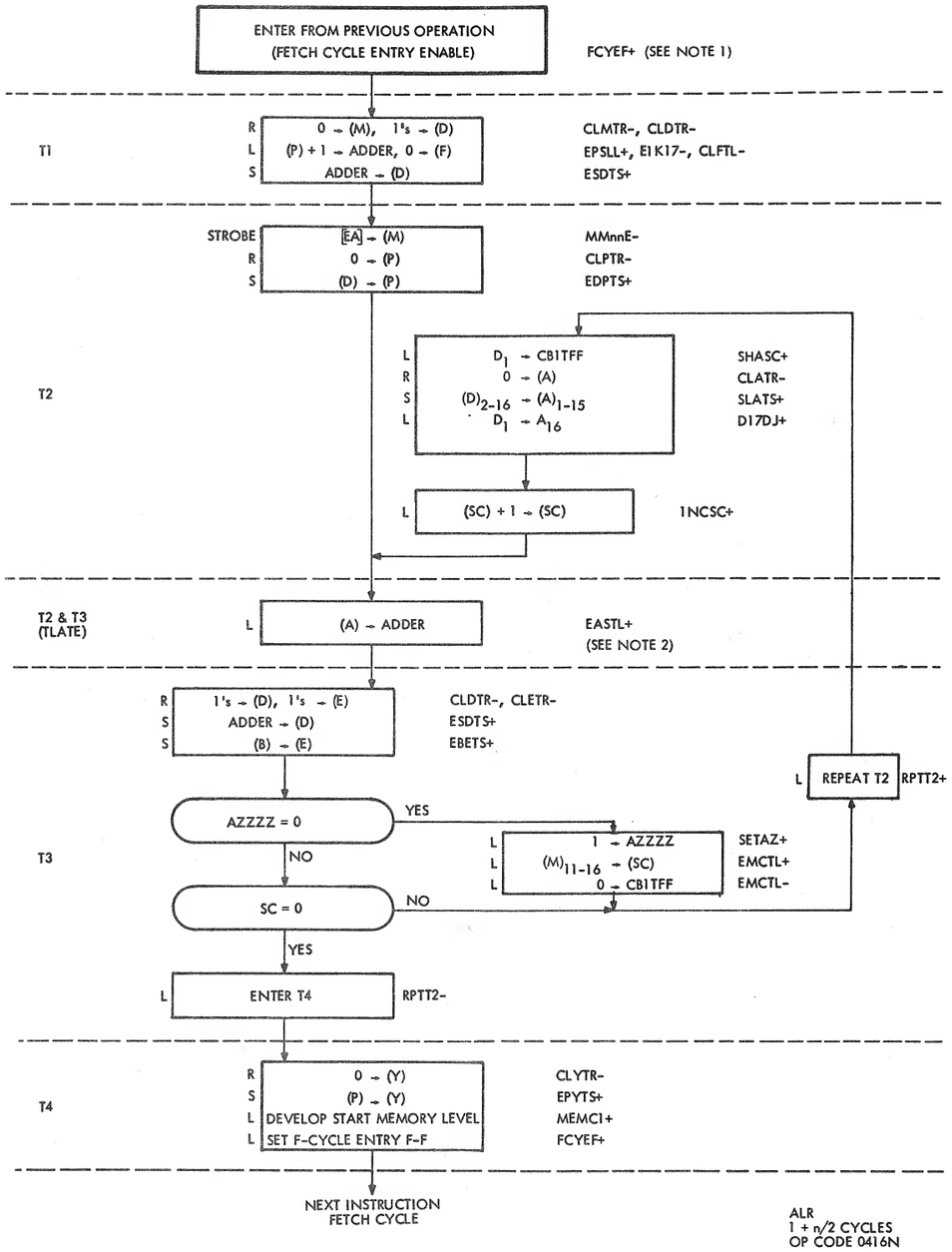

Execution Time ($\mu$sec): 1.92

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| H0LDM- | 128-H8 | A | | | (10RGP+)(ACYEF+) | 128-H8 | 128-K8 | Inhibit M-register clear |
| ADB7-16 | 138-11 | A | | | (DMCCY-)(M__FF+) | 138-11 | (See main frame wire list) | |
| CLDTR- | 125-J5 | A | TL1 | R | (ACYEF+)(TL1FF+)(JST0P-)(1RS0P-)(1MA0P-)(MCRST+) | 125-D4 | 101--116-E7 | Clear D-register to ONEs |
| EPSLL+ | 128-H4 | A | TL1 | L | (10GRP+) | 128-D5 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | A | TL1 | L | (ACYLF+)(CAS0P-)(LSX0P-)(SUB0P-) | 127-J5 | 117-B1 | Force carry to adder |
| ESDTS+ | 125-L4 | A | TL1 | S | (ACYEF+)(TL1FF+)(JST0P-)(1RS0P-)(1MA0P-)(MCSET+) | 125-D4 | 101--116-D4-D8 | Enable adder sum to D-register |
| DRL1N- | 143-B3 | A | | | Function of I/O option | | 125-D10 | Device ready line |
| N0P10- | 125-D10 | A | TL1 | L | (ACYEF+)(TL1FF+)(10GRP+)(DRL1N-) | 125-D10 | 125-L9 | Set AZZZZ FF |
| CLPTR- | 129-J10 | A | TL2 | R | (TL2FF+)(10GRP+)(AZZZZ-)(0CPLS-)(MCRST+) | 129-F8 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | A | TL2 | S | (TL1FF+)(10GRP+)(AZZZZ-)(0CPLS-)(MCSET+) | 129-F8 | 101--116-H11 | Enable D-register to P-register |
| CLYTR- | 129-J3 | A | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-B1/D1 | 101--116-G7 | Clear Y-register |
| EPYTS+ | 129-L5 | A | TL4 | S | (P1SEX-)(E01NS+)(TL4FF+)(0PGJS-)(MCSET+)(DMCRQ-) | 129-D4/H4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | A | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | A | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

Common shift instruction entry

| Function | Origin | Cyc | Tim | Clk | Boolean Expression | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| EPSLL+ | 128-H4 | F | TLATE | L | (FCYEF+)(TLATE-) | 128-F3 | 101--116-A10 | Enable P-register to adder |
| E1K17- | 127-L4 | F | TLATE | L | (TLATE-) | 127-J5 | 116-D7/D9 | Force carry to adder |
| CLFTL- | 125-J7 | F | TL1 | L | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 121-A5<br>120-B1<br>125-L10 | Clear shift counter<br>Clear F-register<br>Clear AZZZZ FF |
| CLMTR- | 128-K8 | F | TL1 | R | (MCRST+)(H0LDM-)(TL1FF+) | 128-K8 | 101--116-H9 | Clear M-register |
| CLDTR- | 125-J5 | F | TL1 | R | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-E7 | Clear D-register to ONEs |
| ESDTS+ | 125-L4 | F | TL1 | S | (1CYEF-)(ACYEF-)(TL1FF+) | 125-A5 | 101--116-D4-D8 | Enable adder sum to D-register |
| MMnnE- | 153/160 | | | | (SWnnA+)(STRB1+) | 153/160 | 101--116-H8 | Memory data set into M-register |
| CLPTR- | 129-J10 | F | TL2 | R | (EDPTL+)(MCRST+)* | 129-H9 | 101--116-H10 | Clear P-register |
| EDPTS+ | 129-L9 | F | TL2 | S | (EDPTL+)(MCSET+) | 129-H9 | 101--116-G11 | Enable D-register to P-register |
| EASTL+ | 127-L1 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G11 | 101--116-A5 | Enable A-register to adder |
| EMSHL+ | 127-L8 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G11 | 101--108-A8 | Enable M(1-7) to adder |
| ENSHL+ | 127-L7 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G11 | 101--107-A9 | Enable M-(1-7) to adder |
| EMSLL+ | 127-L10 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G11 | 108--116-A8 | Enable M(8-16) to adder |
| ENSLL+ | 127-L5 | F | TLATE | L | (GEN0P+)(TLATE+)(M01FF-) | 127-G11 | 108--116-A9 | Enable M-(8-16) to adder |
| SETAZ+ | 125-J9 | F | TL3 | L | (SHA0P+)(TL3FF+)(AZZZZ-) | 125-F11 | 125-L10 | Set AZZZZ FF |
| EMCTL+ | 125-L11 | F | TL3 | L | (SHA0P+)(TL3FF+)(AZZZZ-) | 125-F11 | 121-A11 | Enable M-register to shift counter |
| EMCTL- | 125-J11 | F | TL3 | L | (SHA0P+)(TL3FF+)(AZZZZ-) | 125-F11 | 124-F1 | Clear CB1TF |
| SCZR0- | 121-L3 | | | | (SC16F-) through (SC11F-) | 121-X7 | 126-D4 | Shift counter equals ZERO |
| RPTT2+ | 126-F5 | F | TL3 | L | NOR of EMC16 through EMC11- | 126-F10 | 118-A3 | Repeat TL2 timing level |
| CLDTR- | 125-J5 | F | TL3 | R | (TL3FF+)(ACYLF-)(MCRST+) | 125-D6 | 101--116-E7 | Clear D-register to ONEs |
| CLETR- | 125-J2 | F | TL3 | R | (TL3FF+)(GEN0P+)(M01FF-)(M02FF+)(MCRST+) | 125-D1 | 101--116-L3 | Clear E-register |
| ESDTS+ | 125-L4 | F | TL3 | S | (TL3FF+)(10GRP-)(MCSET+) | 125-D6 | 101--116-D4-D8 | Enable adder sum to D-register |
| EBETS+ | 125-L1 | F | TL3 | S | (TL3FF+)(GEN0P+)(M01FF-)(M02FF+)(MCSET+) | 125-D1 | 101--116-J2 | Enable B-register to E-register |

*See gate A1A44-F at 129-D8 for EDPTL+




NOTES: 1. THIS INSTRUCTION IDENTICAL TO LGL EXCEPT FOR STATE OF D17DJ

2. MISSING SIGNALS CAN BE FOUND IN ALR ENTRY ANALYSIS

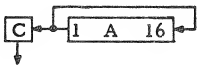
3474

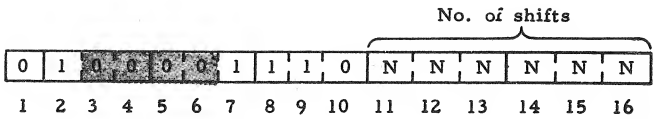
Instruction: Logical Left Rotate (ALR)

OP Code: 0416N

Type: SH, 1 + n/2 cycles
n = no. of shifts

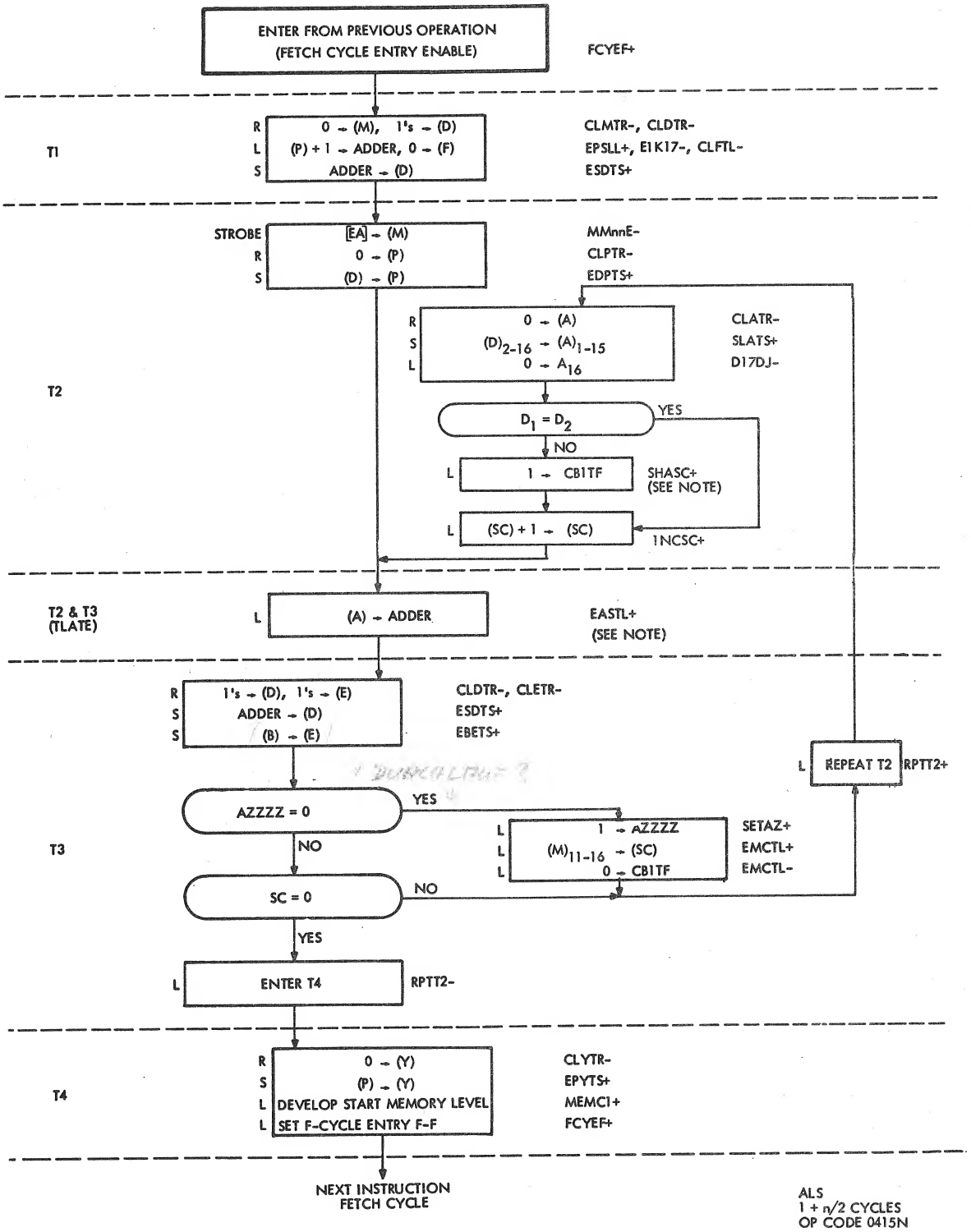
Description:




Execution Time (μsec): 0.96 + 0.48n

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operatior Description |
|---|---|---|---|---|---|---|---|---|
| | | | | | Common Entry (See Common Entry for Shift Instructions, Page 2-90) | | | |
| SHASC+ | 120-K8 | F | TL2 | L | (SHA0P+)(SCZR0-) | 120-H8 | 122-D10<br>124-D1/D3 | Shift A-register and shift counter not equal to ZERO |
| CB1TF | 124-L1 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF+)(M10FF-)(D01FF+)(MCSET+) | 122-D3 | 124-L2 | Set D1 into CB1TF |
| CLATR- | 122-H7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF+)(MCRST+) | 122-D10 | 101--116-H5 | Clear A-register |
| D17DJ+ | 130-F3 | F | TL2 | L | (SHA0P+)(M09FF+)(M08FF+)(D01FF+) | 130-D2 | 116-G5 | Left shift end effect |
| SLATS+ | 122-L10 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF+)(MCSET+) | 124-D3 | 101--116-G5 | Shift left A-register |
| | | | | | Common Exit (See Common Exit for Shift Instructions, Page 2-113) | | | |



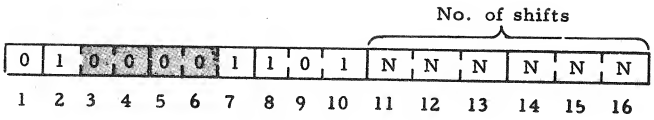

NOTE: MISSING SIGNALS CAN BE FOUND IN ALS ENTRY ANALYSIS

8490

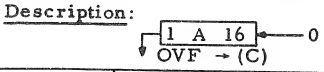
Instruction: Arithmetic Left Shift (ALS)

OP Code: 0415N  Type: SH, 1 + n/2 cycles

n = no. of shifts

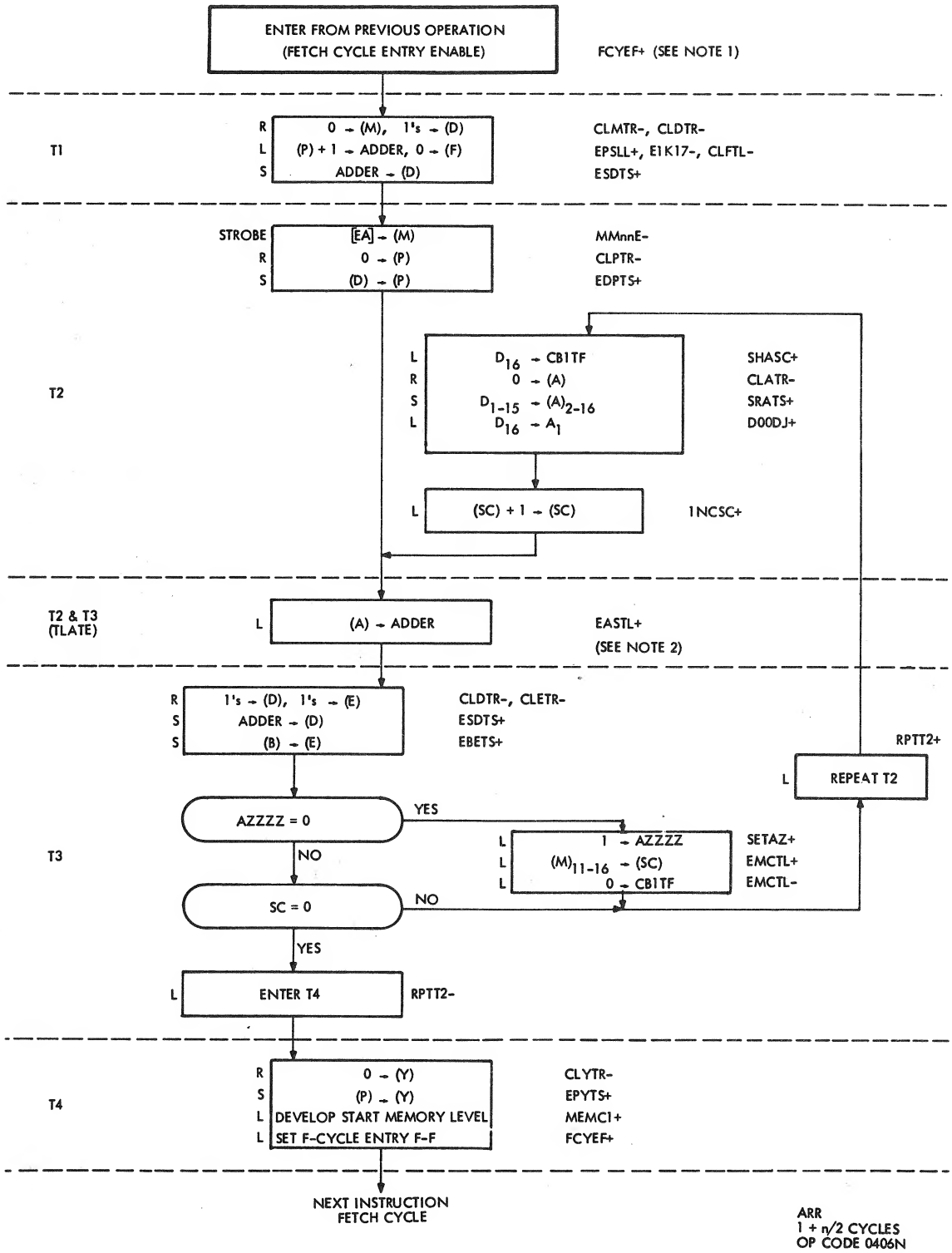



Description:




Execution Time (μsec): 0.96 + 0.48n

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| | | | | | Common Entry (See Common Entry for Shift Instructions, Page 2-90) | | | |
| SHASC+ | 120-K8 | F | TL2 | L | (SHA0P+)(SCZR0-) | 120-H8 | 122-D10<br>124-D7 | Shift A-register and shift counter not equal to ZERO |
| CLATR- | 122-H7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF+)(MCRST+) | 122-C10 | 101--116-H5 | Clear A-register |
| D17DJ- | 130-F4 | F | TL2 | L | (M08FF+) | 130-D4 | 116-G5 | Clear A-register bit 16 |
| SLATS+ | 122-L10 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF+)(MCSET+) | 122-D10 | 101--116-G5 | Shift left A-register |
| D1≠D2 | 124-D7 | F | TL2 | L | See inputs to gate A1E37-F | 124-D7 | 124-F2 | Set CB1TF with SHASC+ |
| | | | | | Common Exit (See Common Exit for Shift Instructions, Page 2-113) | | | |




NOTES: 1. THIS INSTRUCTION IDENTICAL TO LGR EXCEPT FOR STATE OF D00DJ

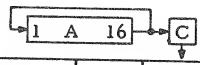
2. MISSING SIGNALS CAN BE FOUND IN ARR ENTRY ANALYSIS

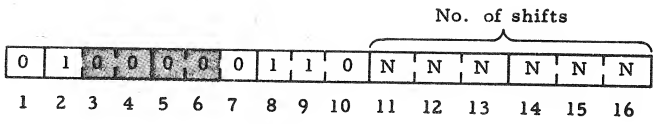
3472

Instruction: Logical Right Rotate (ARR)

OP Code: 0406N Type: SH, 1+ n/2 cycles

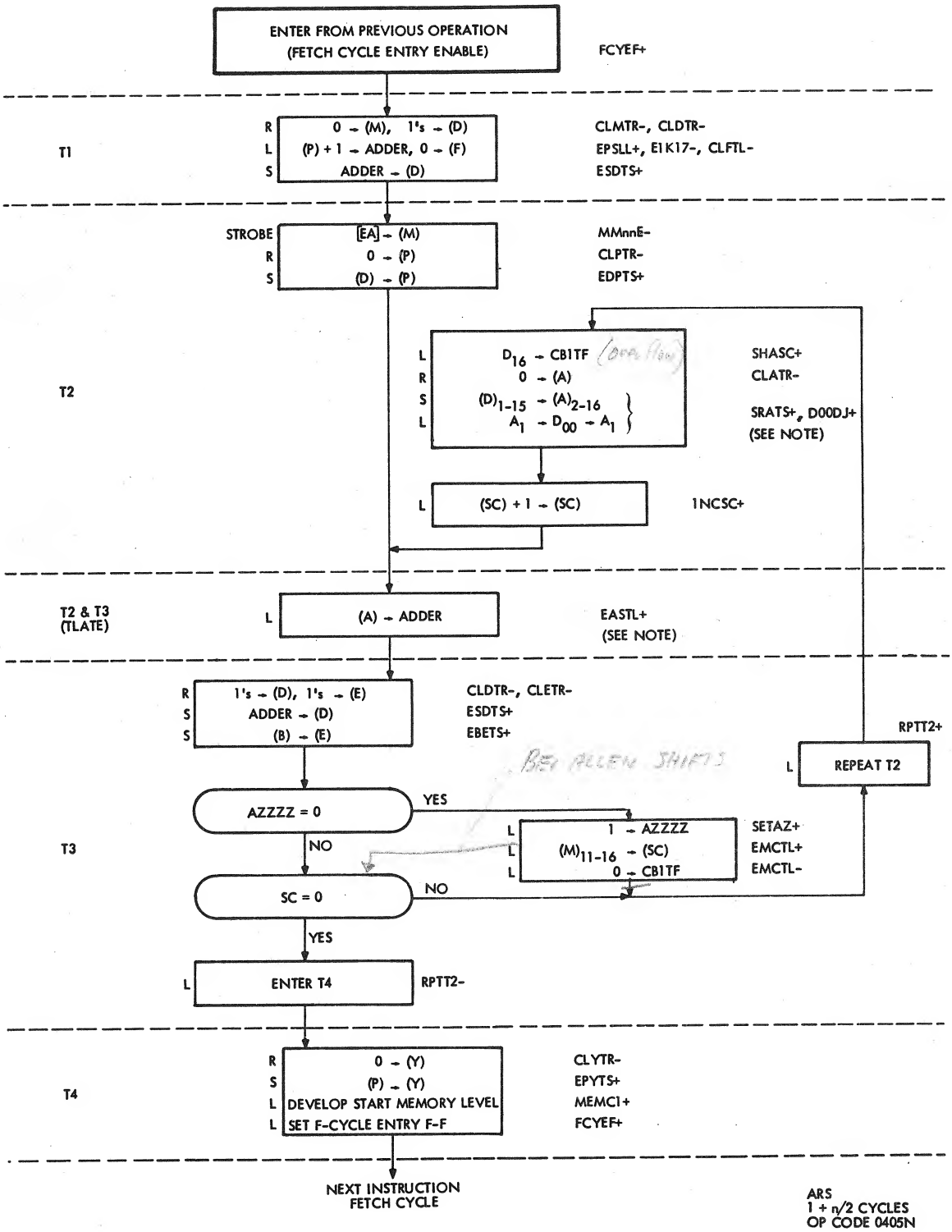
Description: n = no. of shifts




Execution Time (μsec): 0.96 + 0.48n

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| | | | | | Common Entry (See Common Entry for Shift Instructions, Page 2-90) | | | |
| SHASC+ | 120-K8 | F | TL2 | L | (SHA0P+)(SCZR0-) | 120-H8 | 122-D11<br>124-D1/D4 | Shift A-register and shift counter not equal to ZERO |
| CB1TF | 124-L1 | F | TL2 | S | (SHASC+)(M07FF-)(M08FF+)(TL2FF+)(D16FF+)(MCSET+) | 124-D4 | 124-L2 | Set D16 into CB1TF |
| CLATR- | 122-H7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF-)(MCRST+) | 122-D11 | 101--116-H5 | Clear A-register |
| D00DJ+ | 130-D1 | F | TL2 | L | (SHA0P+)(M02FF+)(M09FF+)(D16FF+) | 130-B2 | 101-G6 | Right shift end effect |
| SRATS+ | 122-L11 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF-)(MCSET+) | 122-D11 | 101--116-G6 | Shift right A-register |
| MFGZE- | 122-F1 | F | TL2 | L | (SHASC+)(TL2FF+)(M07FF-) | 122-D11 | 124-F1 | Clear CB1TF |
| | | | | | Common Exit (See Common Exit for Shift Instructions, Page 2-113) | | | |



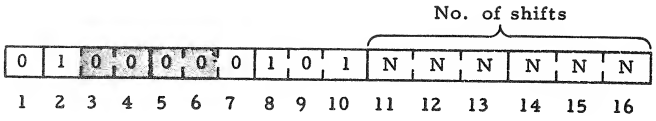

NOTE: MISSING SIGNALS CAN BE FOUND IN ARS ENTRY ANALYSIS

3492

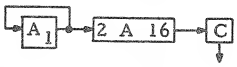
Instruction: Arithmetic Right Shift (ARS)

OP Code: 0405N Type: SH, 1 + n/2 cycles

n = no. of shifts

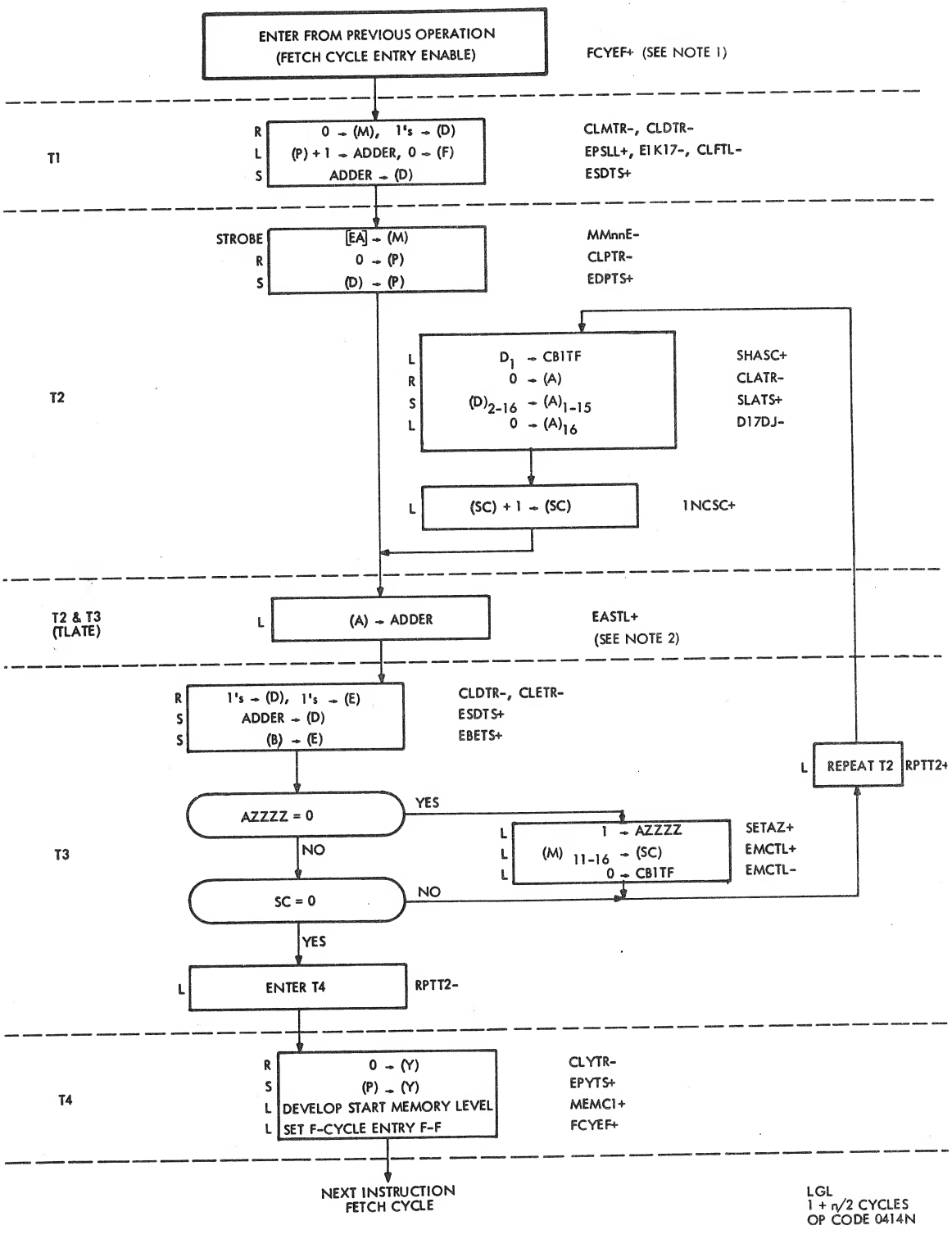



Description:




Execution Time (μsec): 0.96 + 0.48n

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| | | | Common Entry (See Common Entry for Shift Instructions, Page 2-90) | | | | | |
| SHASC+ | 120-K8 | F | TL2 | L | (SHA0P+)(SCZR0-) | 120-H8 | 122-D11<br>124-D4 | Shift A-register and shift counter not equal to ZERO |
| CB1TF | 124-L1 | F | TL2 | S | (SHASC+)(M07FF-)(M08FF+)(TL2FF+)(D16FF+)(MCSET+) | 124-D4 | 124-L2 | Set D16 into CB1TF |
| MFG2E- | 122-F1 | F | TL2 | L | (SHASC+)(M07FF-)(TL2FF+) | 122-D11 | 124-F1 | Clear CB1TF |
| CLATR- | 122-H7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF-)(MCRST+) | 122-D11 | 101--116-H5 | Clear A-register |
| SRATS+ | 122-L1 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF-)(MCSET+) | 122-D11 | 101--116-G6 | Shift right A-register |
| D00DJ+ | 130-D1 | F | TL2 | L | (M10FF+)(D00FF+) | 122-B1 | 101-G6 | Maintain sign bit |
| D00FF+ | 130-H9 | F | TL2 | S | (ESDTS+)(R01PA+)(G01DJ-) | 130-F6 | 130-B1 | Extension of D-register |
| R01PA+ | 101-D9 | F | TL2 | L | (H01DJ-) | 101-B9 | 101-B9 | Adder network OR gate |
| G01DJ- | 101-A4 | F | TL2 | L | (EASTL+)(A01FF+) | 101-A4 | 101-A4 | Adder network OR gate |
| | | | Common Exit (See Common Exit for Shift Instructions, Page 2-113) | | | | | |




NOTES: 1. THIS INSTRUCTION IDENTICAL TO ALR EXCEPT FOR STATE OF D17DJ

2. MISSING SIGNALS CAN BE FOUND IN LGL ENTRY ANALYSIS

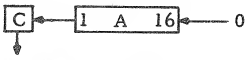
3473

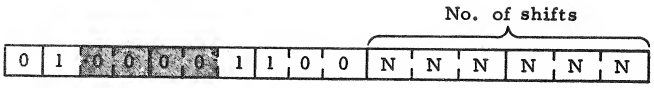
Instruction: Logical Left Shift (LGL)

OP Code: 0414N Type: SH, 1 + n/2 cycles

n = no. of shifts

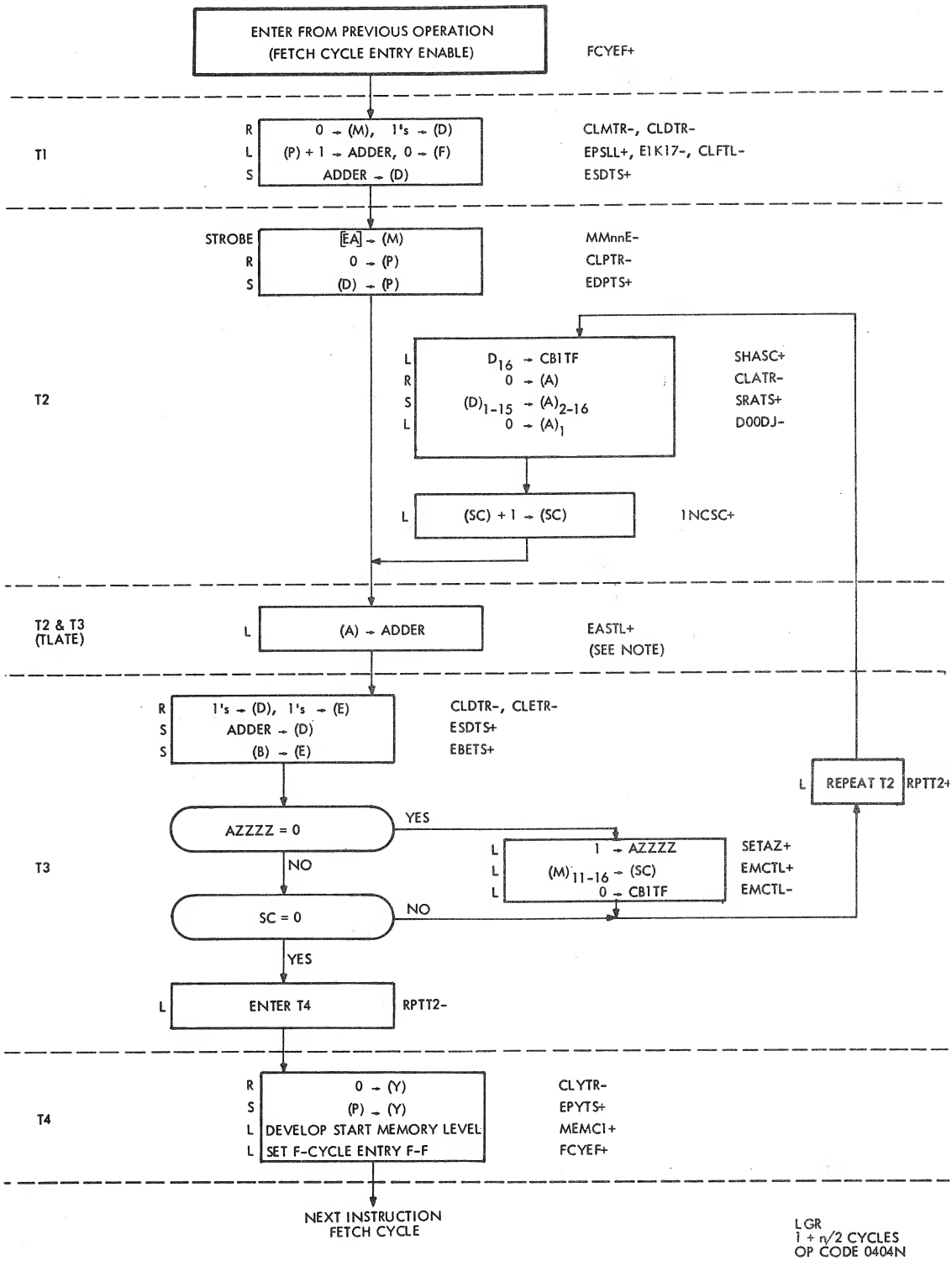
Description:




Execution Time (μsec): 0.96 + 0.48n

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| | | | | | Common Entry (See Common Entry for Shift Instructions, Page 2-90) | | | |
| SHASC+ | 120-K8 | F | TL2 | L | (SHA0P+)(SCZR0-) | 120-H8 | 122-D10<br>124-D1 | Shift A-register and shift counter not equal to ZERO |
| CB1TF | 124-L1 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF+)(M10FF-)(D01FF+)(MCSET+) | 122-D3 | 124-L2 | Set D1 into CB1TF |
| CLATR- | 122-H7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF+)(MCRST+) | 122-D10 | 101--116-H5 | Clear A-register |
| D17DJ- | 130-F4 | F | TL2 | L | (M08FF+) | 130-D3 | 116-G5 | Clear A-register bit 16 |
| SLATS+ | 122-L10 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF+)(MCSET+) | 122-D10 | 101--116-G5 | Shift left A-register |
| | | | | | Common Exit (See Common Exit for Shift Instructions, Page 2-113) | | | |




NOTES: 1. THIS INSTRUCTION IDENTICAL TO ARR EXCEPT FOR STATE OF D00DJ

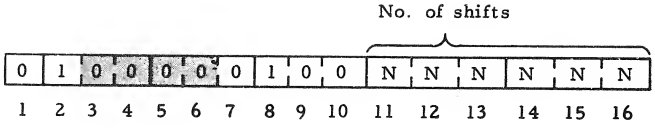
2. MISSING SIGNALS CAN BE FOUND IN LGR ENTRY ANALYSIS

3478

Instruction: Logical Right Shift (LGR)

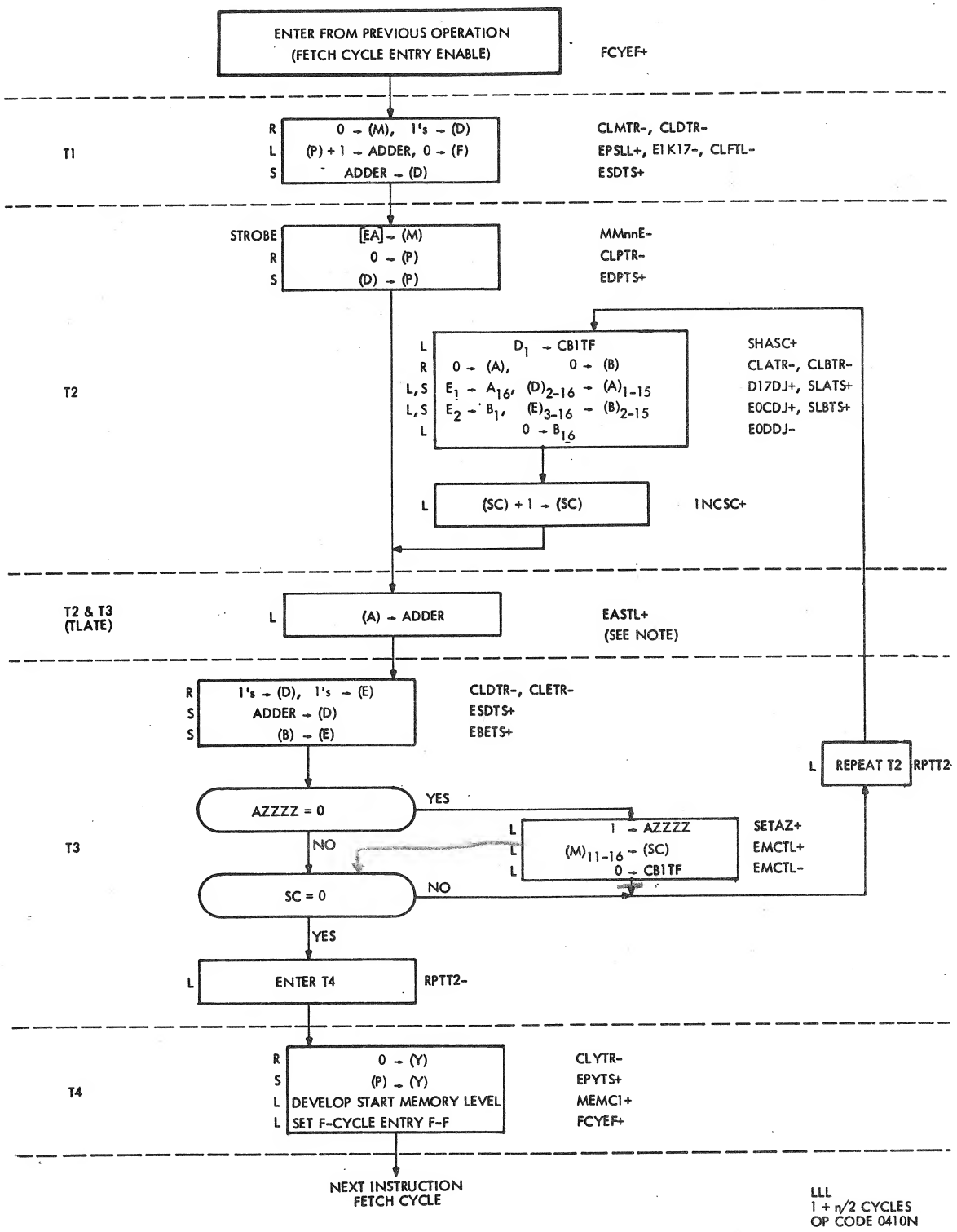
OP Code: 0404N  Type: SH, 1 + n/2 cycles
n = no. of shifts

Description: 0 → [1 A 16] → [C]




Execution Time (μsec): 0.96 + 0.48n

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| | | | Common Entry (See Common Entry for Shift Instructions, Page 2-90) | | | | | |
| SHASC+ | 120-K8 | F | TL2 | L | (SHA0P+)(SCZR0-) | 120-H8 | 122-D11<br>124-D1/D4 | Shift A-register and shift counter not equal to ZERO |
| CB1TF | 124-L1 | F | TL2 | S | (SHASC+)(M07FF-)(M08FF+)(TL2FF+)(D16FF+)(MCSET+) | 124-D4 | 124-L2 | Set D16 into CB1TF |
| CLATR- | 122-H7 | F | TL2 | R | (SHASC+)(M07FF-)(TL2FF+)(MCRST+) | 122-D11 | 101--116-H5 | Clear A-register |
| SRATS+ | 122-L11 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF-)(MCSET+) | 122-D11 | 101--116-G6 | Shift right A-register |
| MFG2E- | 122-F1 | F | TL2 | L | (SHASC+)(TL2FF+)(M07FF-) | 122-D11 | 124-F1 | Clear CB1TF |
| D00DJ- | 130-D1 | F | TL2 | L | (M10FF-)(M08FF+) | 130-B3 | 101-G6 | Clear A-register bit 1 |
| | | | Common Exit (See Common Exit for Shift Instructions, Page 2-113) | | | | | |




NOTES: 1. THIS INSTRUCTION IDENTICAL TO LLR EXCEPT FOR STATE OF E0DDJ

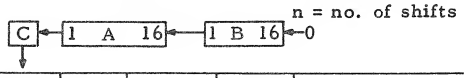
2. MISSING SIGNALS CAN BE FOUND IN LLL ENTRY ANALYSIS

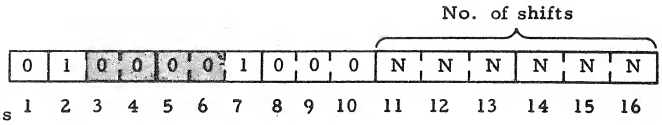
3477

Instruction: Long Left Logical Shift (LLL)

OP Code: 0410 Type: SH, 1 + n/2 cycles

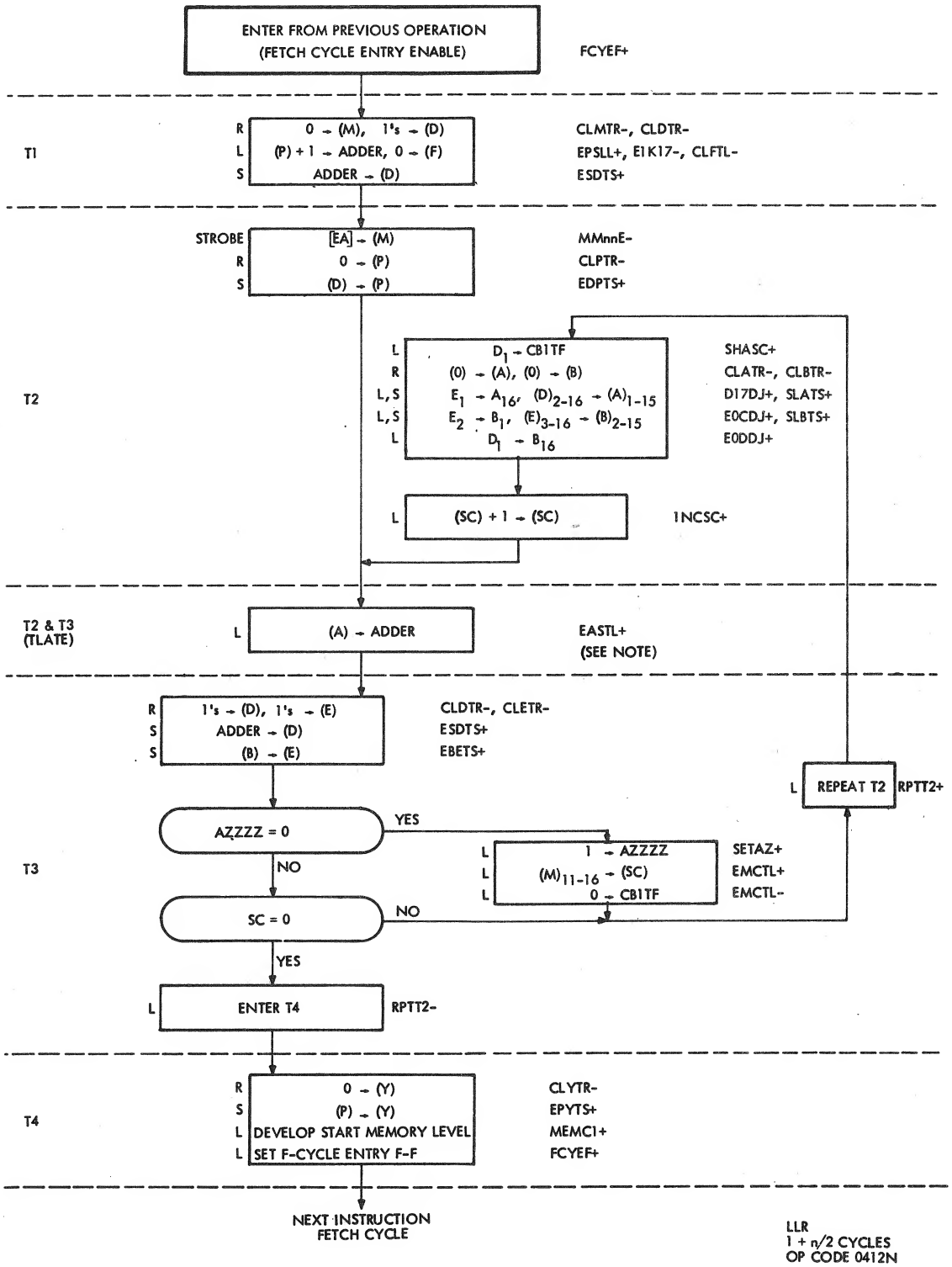
Description:




Execution Time (μsec): 0.96 + 0.48n

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| | | | | | Common Entry (See Common Entry for Shift Instructions, Page 2-90) | | | |
| SHASC+ | 120-K8 | F | TL2 | L | (SHA0P+)(SCZR0-) | 120-H8 | 122-C10<br>123-E4<br>124-D1/D3 | Shift A-register and shift counter not equal to ZERO |
| CB1TF | 124-L1 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF+)(M10FF-)(D01FF+)(MCSET+) | 122-D3 | 124-L2 | Set D1 into CB1TF |
| CLATR- | 122-H7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF+)(MCRST+) | 122-C10 | 101--116-H5 | Clear A-register |
| CLBTR- | 123-J7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF+)(M08FF-)(MCRST+) | 123-E4 | 101--116-H2 | Clear B-register |
| D17DJ+ | 130-F3 | F | TL2 | L | (ACYLF-)(M08FF-)(M10FF-)(E01FF+) | 130-D3 | 116-G5 | Left shift end effect |
| SLATS+ | 122-L10 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF+) | 122-D10 | 101--116-G5 | Shift left A-register |
| E0CDJ+ | 130-D8 | F | TL2 | L | (E02FF+) | 130-D8 | 101-E1 | Set E2 into B1 |
| SLBTS+ | 123-L4 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF+)(M08FF-)(MCSET+) | 123-E4 | 101--116-E1 | Shift left B-register |
| E0DDJ- | 130-D11 | F | TL2 | L | (M09FF-) | 130-B11 | 116-G1 | Clear B16 |
| | | | | | Common Exit (See Common Exit for Shift Instructions, Page 2-113) | | | |




NOTES: 1. THIS INSTRUCTION IDENTICAL TO LLL EXCEPT FOR STATE OF E0DDJ

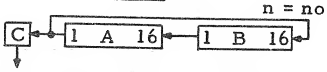
2. MISSING SIGNALS CAN BE FOUND IN LLR ENTRY ANALYSIS

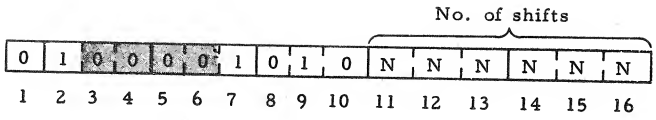
3479

Instruction: Long Left Rotate (LLR)

OP Code: 0412N  Type: SH, 1 + n/2 cycles

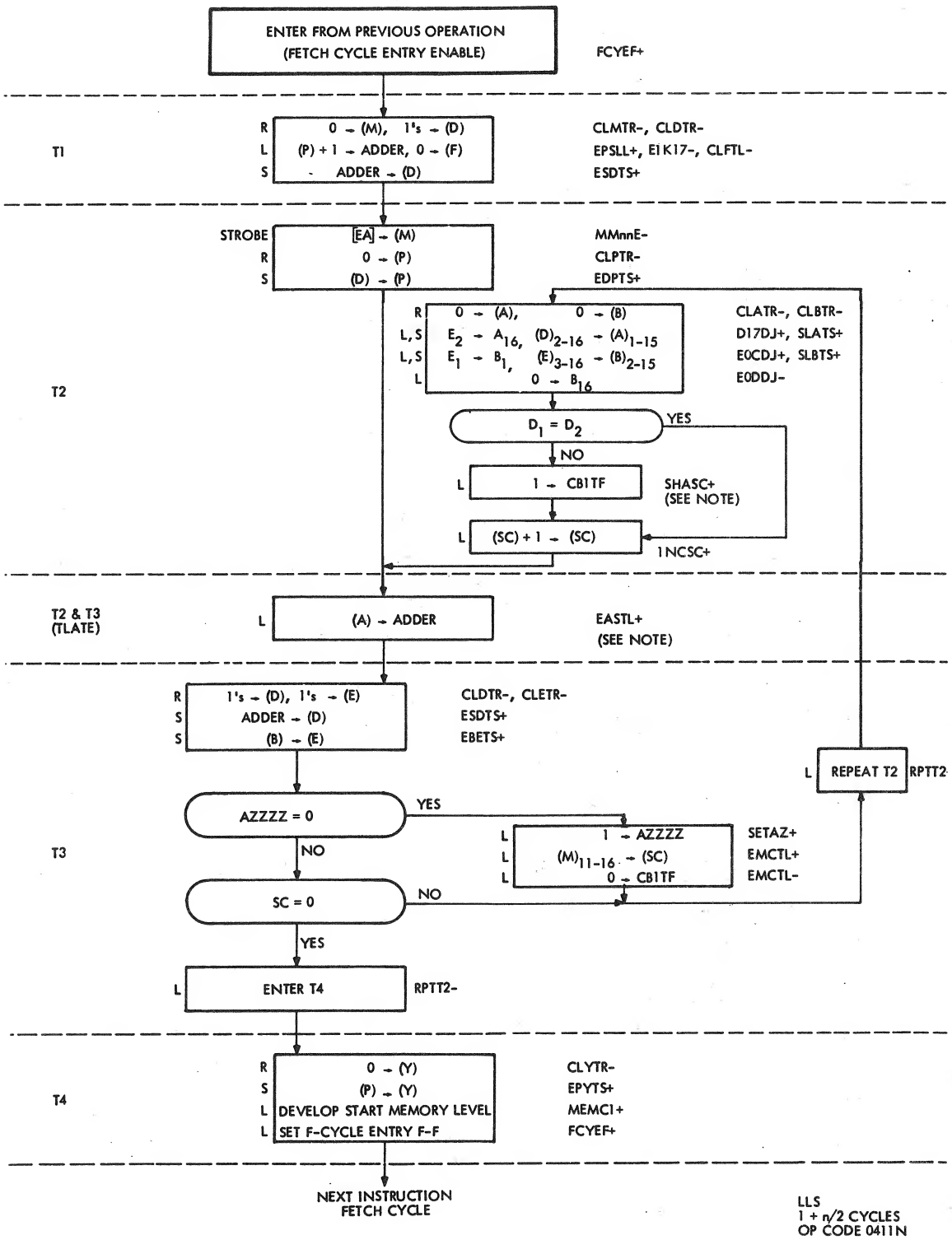
Description: n = no. of shifts




Execution Time (μsec): 0.96 + 0.48n

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| | | | | | Common Entry (See Common Entry for Shift Instructions, Page 2-90) | | | |
| SHASC+ | 120-K8 | F | TL2 | L | (SHA0P+)(SCZR0-) | 120-H8 | 122-C10<br>123-E4<br>124-C1/C3 | Shift A-register and shift counter not equal to ZERO |
| CB1TF | 124-L1 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF+)(M10FF-)(D01FF+)(MCSET+) | 122-D3 | 124-L2 | Set D1 into CB1TF |
| CLATR- | 122-H7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF+)(MCRST+) | 122-C10 | 101--116-H5 | Clear A-register |
| CLBTR- | 123-J7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF+)(M08FF-)(MCRST+) | 123-E4 | 101--116-H2 | Clear B-register |
| D17DJ+ | 130-F3 | F | TL2 | L | (ACYLF-)(M08FF-)(M10FF-)(E01FF+) | 130-D3 | 116-G5 | Left shift end effect |
| SLATS+ | 122-L10 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF+)(MCSET+) | 122-D10 | 101--116-G5 | Shift left A-register |
| E0CDJ+ | 130-D8 | F | TL2 | L | (E02FF+) | 130-D8 | 101-E1 | Set E2 and B1 |
| SLBTS+ | 123-L4 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF+)(M08FF-)(MCSET+) | 123-E4 | 101--116-E1 | Shift left B-register |
| E0DDJ+ | 130-D11 | F | TL2 | L | (SHA0P+)(M09FF+)(D01FF+) | 130-B11 | 116-G1 | Set D1 into B16 |
| | | | | | Common Exit (See Common Exit for Shift Instructions, Page 2-113) | | | |



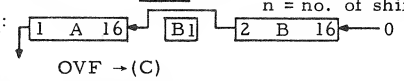

NOTE: MISSING SIGNALS CAN BE FOUND IN LLS ENTRY ANALYSIS

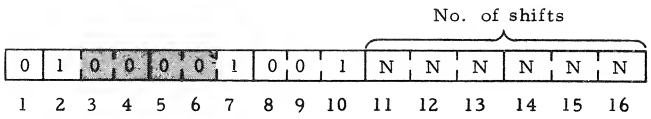
3491

Instruction: Long Arithmetic Left Shift (LLS)

OP Code: 0411N Type: SH, 1 + n/2 cycles
n = no. of shifts

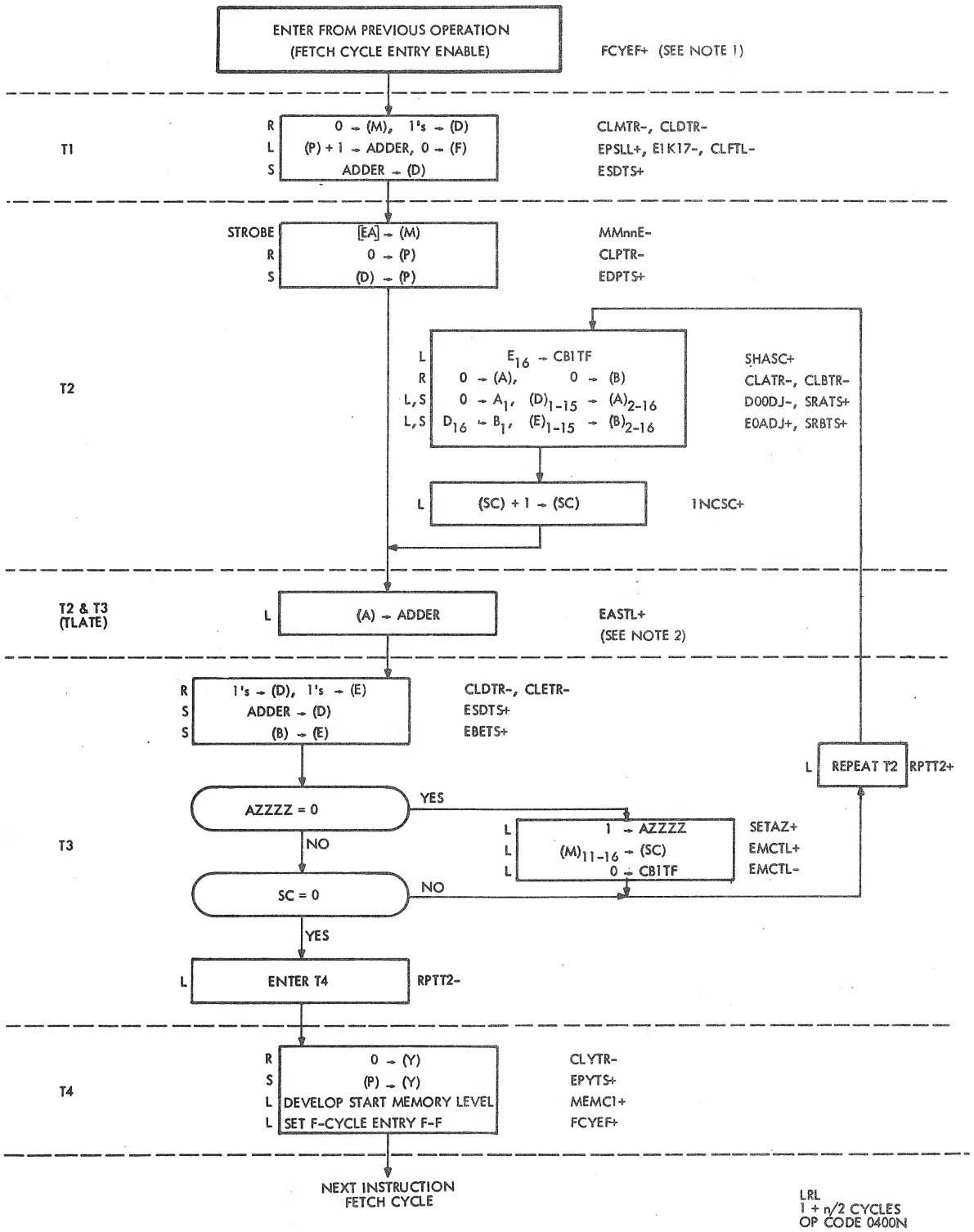
Description:




Execution Time (μsec): 0.96 + 0.48n

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| | | | | | Common Entry (See Common Entry for Shift Instructions, Page 2-90) | | | |
| SHASC+ | 120-K8 | F | TL2 | L | (SHA0P+)(SCZR0-) | 120-H8 | 122-D10<br>123-E4<br>124-D7 | Shift A-register and shift counter not equal to ZERO |
| CLATR- | 122-H7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF+)(MCRST+) | 122-C10 | 101--116-H5 | Clear A-register |
| CLBTR- | 123-J7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF+)(M08FF-)(MCRST+) | 123-E4 | 101--116-H2 | Clear B-register |
| D17DJ+ | 130-F3 | F | TL2 | L | (M08FF-)(M10FF+)(E02FF+) | 130-D4 | 116-G5 | Left shift end effect |
| SLATS+ | 122-L10 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF+)(MCSET+) | 122-D10 | 101--116-G5 | Shift left A-register |
| E0CDJ+ | 130-D8 | F | TL2 | L | (ACYLF-)(M10FF+)(E01FF-) | 130-D7 | 101-E1 | Set E1 into B1 |
| SLBTS+ | 123-L4 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF+)(M08FF-)(MCSET+) | 123-E4 | 101--116-E1 | Shift left B-register |
| E0DDJ- | 130-D11 | F | TL2 | L | (M09FF-) | 130-B11 | 116-G1 | Clear B16 |
| D1 ≠ D2 | 124-D7 | F | TL2 | L | See inputs to gate A1E37-F | 124-D7 | 124-F2 | Set CB1T with SHASC+ |
| | | | | | Common Exit (See Common Exit for Shift Instructions, Page 2-113) | | | |



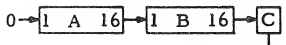

NOTES:
1. THIS INSTRUCTION IDENTICAL TO LRR EXCEPT FOR STATE OF D00DJ
2. MISSING SIGNALS CAN BE BE FOUND IN LRL ENTRY ANALYSIS

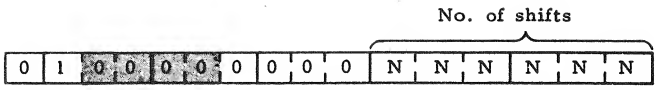
3478

Instruction: Long Right Logical Shift (LRL)

OP Code: 0400N  Type: SH, 1 + n/2 cycle
n = no. of shifts

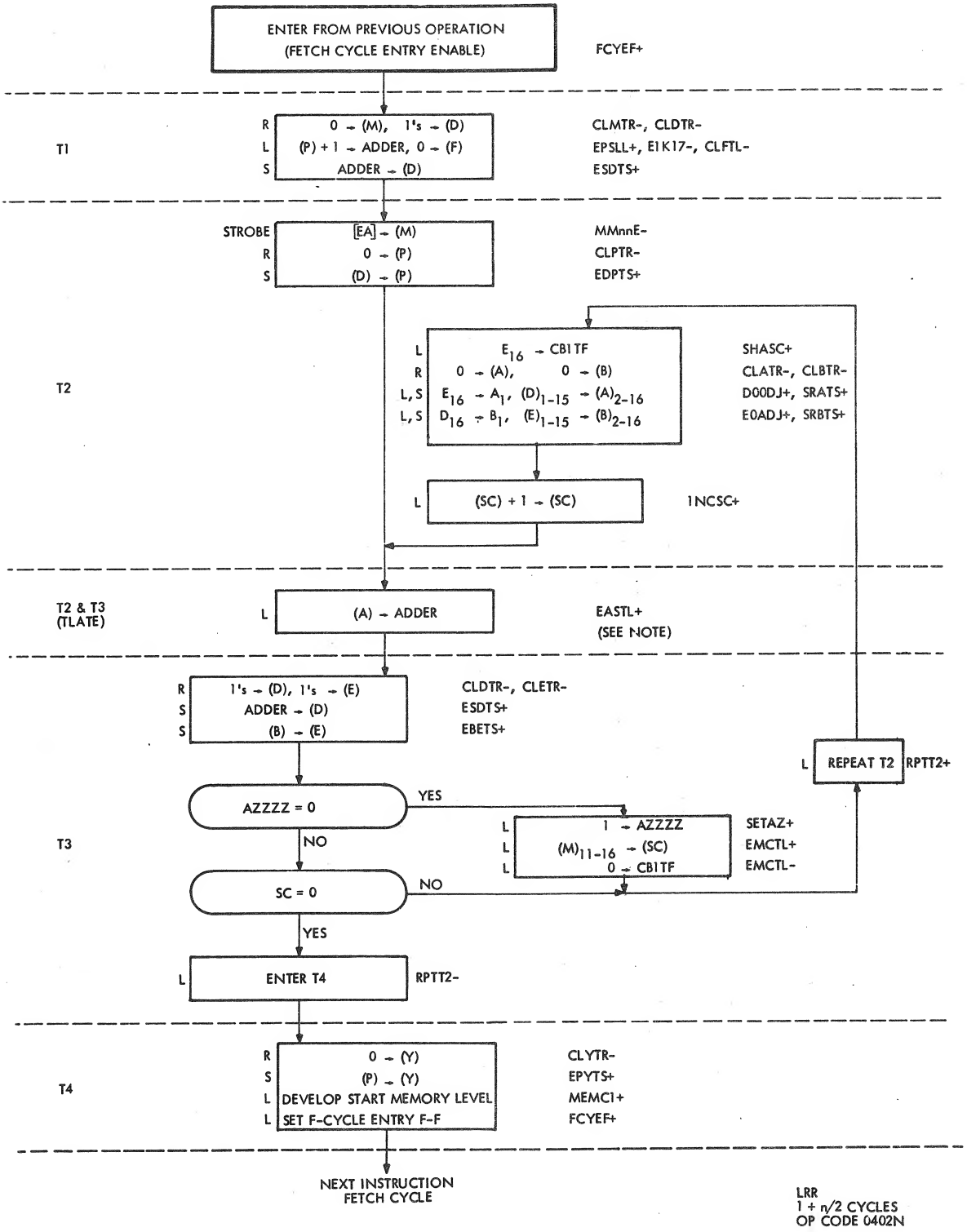
Description:




Execution Time (μsec): 0.96 + 0.48n

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| | | | | | Common Entry (See Common Entry for Shift Instructions, Page 2-90) | | | |
| SHASC+ | 120-K8 | F | TL2 | L | (SHA0P+)(SCZR0-) | 120-H8 | 122-C11<br>123-E6<br>124-D1/D5 | Shift A-register and shift counter not equal to ZERO |
| CB1TF | 124-L1 | F | TL2 | S | (TL2FF+)(SHASC+)(M07FF-)(M08FF-)(E16FF+)(MCSET+) | 124-D5 | 124-L2 | Set E16 into CB1TF |
| MFG2E- | 122-F1 | F | TL2 | L | (SHASC+)(TL2FF+)(M07FF-) | 122-D11 | 124-F1 | Clear CB1TF |
| CLATR- | 122-H7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF-)(MCRST+) | 122-C11 | 101--116-H5 | Clear A-register |
| CLBTR- | 123-J7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF-)(M08FF-)(MCRST+) | 123-E6 | 101--116-H2 | Clear B-register |
| D00DJ- | 130-D1 | F | TL2 | L | (M10FF-)(M09FF-) | 130-B2 | 101-G6 | Clear A-register bit 1 with SRATS+ |
| SRATS+ | 122-L1 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF-)(MCSET+) | 122-D11 | 101--116-G6 | Shift right A-register |
| E0ADJ+ | 130-D9 | F | TL2 | L | (M10FF-)(D16FF-) | 130-D10 | 101-E2 | Set D16 into B1 |
| SRBTS+ | 123-L8 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF-)(M08FF-)(MCSET+) | 123-E6 | 101--116-E2 | Shift right B-register |
| | | | | | Common Exit (See Common Exit for Shift Instructions, Page 2-113) | | | |




NOTES: 1. THIS INSTRUCTION IDENTICAL TO LRL EXCEPT FOR STATE OF D00DJ

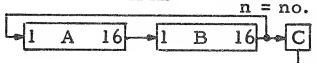
2. MISSING SIGNALS CAN BE FOUND IN LRR ENTRY ANALYSIS

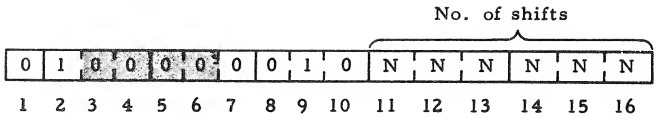
3476

**Instruction:** Long Right Rotate (LRR)

**OP Code:** 0402N **Type:** SH, 1 + n/2 cycles

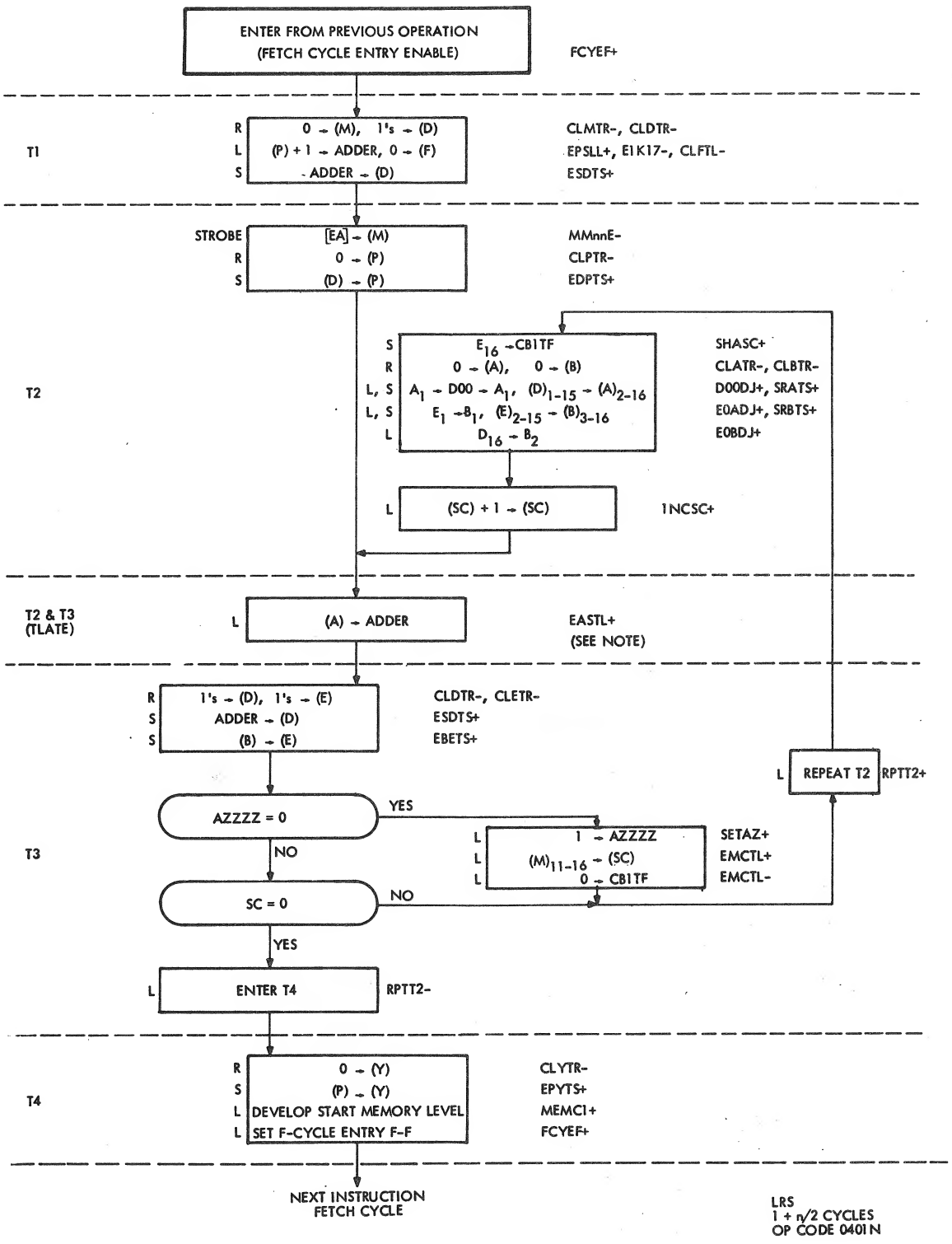
**Description:** n = no. of shifts




**Execution Time (μsec):** 0.96 + 0.48n

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| | | | Common Entry (See Common Entry for Shift Instructions, Page 2-90) | | | | | |
| SHASC+ | 120-K8 | F | TL2 | L | (SHA0P+)(SCZR0-) | 120-H8 | 122-C11<br>123-E6<br>124-D1/D5 | Shift A-register and shift counter not equal to ZERO |
| CB1TF | 124-L1 | F | TL2 | S | (TL2FF+)(SHASC+)(M07FF-)(M08FF-)(E16FF+)(MCSET+) | 124-D5 | 124-L2 | Set E16 into CB1TF |
| CLATR- | 122-H7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF-)(MCRST+) | 122-C11 | 101--116-H5 | Clear A-register |
| CLBTR- | 123-J7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF-)(M08FF-)(MCRST+) | 123-E6 | 101--116-H2 | Clear B-register |
| D00DJ+ | 130-D1 | F | TL2 | L | (SHA0P+)(M08FF-)(M09FF+)(E16FF+) | 130-B3 | 101-G6 | Set E16 into A1 with SRATS+ |
| SRATS+ | 122-L1 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF-)(MCSET+) | 122-D11 | 101--116-G6 | Shift right A-register |
| MFG2E- | 122-F1 | F | TL2 | L | (SHASC+)(TL2FF+)(M07FF-) | 122-D11 | 124-F1 | Clear CB1TF |
| E0ADJ+ | 130-D9 | F | TL2 | L | (M10FF-)(D16FF-) | 130-D10 | 101-E2 | Set D16 into B1 |
| SRBTS+ | 123-L8 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF-)(M08FF-)(MCSET+) | 123-E6 | 101--116-E2 | Shift right B-register |
| | | | Common Exit (See Common Exit for Shift Instructions, Page 2-113) | | | | | |



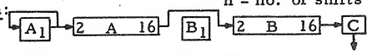

NOTE: MISSING SIGNALS CAN BE FOUND IN LRS ENTRY ANALYSIS

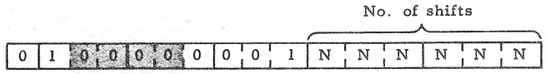
3468

**Instruction:** Long Arithmetic Right Shift (LRS)

**OP Code:** 0401N **Type:** SH, 1 + n/2 cycles
n = no. of shifts

**Description:**




Execution Time ($\mu$sec): 0.96 + 0.48n

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| | | | Common Entry (See Common Entry for Shift Instructions, Page 2-90) | | | | | |
| SHASC+ | 120-K8 | F | TL2 | L | (SHA0P+)(SCZR0-) | 120-H8 | 122-D11<br>123-E7<br>124-D5 | Shift A-register and shift counter not equal to ZERO |
| CB1TF | 124-L1 | F | TL2 | S | (TL2FF+)(SHASC+)(M07FF-)(M08FF-)(E16FF+)(MCSET+) | 124-D5 | 124-L2 | Set E16 into CB1TF |
| CLATR- | 122-H7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF-)(MCRST+) | 122-D11 | 101--116-H5 | Clear A-register |
| CLBTR- | 123-J7 | F | TL2 | R | (SHASC+)(TL2FF+)(M07FF-)(M08FF-)(MCRST+) | 123-E6 | 101--116-H2 | Clear B-register |
| D00DJ+ | 130-D1 | F | TL2 | L | (M10FF+)(D00FF+) | 130-B1 | 101--G6 | Maintain sign bit |
| D00FF+ | 130-H9 | F | TL2 | S | (ESDTS+)(R01PA+)(G01DJ-) | 130-F6 | 130-B1 | Extension of D-register |
| R01PA+ | 101-D9 | F | TL2 | L | (H0LDJ-) | 101-B9 | 101-B9 | Adder network OR gate |
| G01DJ- | 101-A4 | F | TL2 | L | (EASTL+)(A01FF+) | 101-A4 | 101-A4 | Adder network OR gate |
| SRATS+ | 122-L1 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF-)(MCSET+) | 122-D11 | 101--116-G6 | Shift right A-register |
| MFG2E- | 122-F1 | F | TL2 | L | (SHASC+)(TL2FF+)(M07FF-) | 122-D11 | 124-F1 | Clear CB1TF |
| E0ADJ+ | 130-D9 | F | TL2 | L | (M10FF+)(E01FF-) | 130-D9 | 101-E2 | Set E1 into B1 |
| E0BDJ+ | 130-F1 | F | TL2 | L | (M10FF+)(D16FF-) | 130-F1 | 102-F2 | Set D16 into B2 |
| SRBTS+ | 123-L8 | F | TL2 | S | (SHASC+)(TL2FF+)(M07FF-)(M08FF-)(MCSET+) | 123-E6 | 101--116-E2 | Shift right B-register |
| | | | Common Exit (See Common Exit for Shift Instructions, Page 2-113) | | | | | |

Common shift instruction exit

**Instruction:**

| Signal | Origin | Cyc | Tim | Clk | Signal Component | Origin | Destination | Operation Description |
|---|---|---|---|---|---|---|---|---|
| INCSC+ | 126-L3 | F | TL2 | L | (FCYEF+)(TL2FF+)* | 126-H3 | 121-A5 | Increment shift counter |
| CLYTR- | 129-J3 | F | TL4 | R | (ACYNX-)(TL4FF+)(MCRST+) | 129-D3 | 101--116-L11 | Clear Y-register |
| EPYTS+ | 129-L4 | F | TL4 | S | (P1SEX-)(E01NS+)(0PGJS-) | 129-D4/H4 | 101--116-K11 | Enable P-register to Y-register |
| MEMC1+ | 126-J11 | F | TL4 | L | (TL4FF+)(SPM0D-)(TLAFF-) | 126-F11/H11 | 150-C1 | Enable set RCYF1+ |
| RCYF1+ | 150-D1 | F | TL4 | S | (MCSET+)(MEMC1+)(RCYF1-) | 150-C2 | 150-D1 | Start memory cycle |

*Repeat T3 and T2 as required and enter T4

# Honeywell

**COMPUTER CONTROL DIVISION**

## United Kingdom

Honeywell Ltd.,
Computer Control Division,
53 Clarendon Road,
Watford, Herts.
☎ Watford 42391
Telex: 934227

Honeywell Ltd.
Computer Control Division,
Honeywell House,
Station Road,
Cheadle Hulme,
Cheshire.
☎ 061-485 6116
Telex: 66509

Honeywell Ltd.
Computer Control Division,
Observatory House,
Windsor Road,
Slough, Bucks.
☎ Slough 33366
Telex: 84601

## Sweden

Honeywell AB,
Computer Control Division,
Storsätragrand 5,
127 86 Skärholmen,
☎ 08/88 00 00,
Telex: 854-10673

## Finland

OY Honeywell,
Hitsaajankatu 5,
Helsinki 81,
☎ 780311
Telex: 12–1229

## France

Honeywell S.A.,
Computer Control Division,
92 rue de Courcelles,
Paris 8eme.
☎ 227 60 71
Telex: 29623

## Italy

Honeywell S.p.A.,
Computer Control Division,
Via V. Pisani, 13
I 20124 Milano.
☎ 6245
Telex: 32092

## Germany

Honeywell GmbH,
Computer Control Division,
6050 Offenbach/Main,
Kaiserleistrasse 55.
☎ 8.0641
Telex: 41–52758

Honeywell GmbH
Computer Control Division
4 Düsseldorf
Moersenbroicher Weg 200
☎ 6 21 61
Telex: 858 6754

## Netherlands

Honeywell NV,
Computer Control Division,
Egelenburg 150-152,
Amsterdam Buitenveldert.
☎ 020-429666
Telex: 13066

## Switzerland

Honeywell AG,
Computer Control Division,
8008 Zürich,
Dufourstrasse 47.
☎ (051) 47.44.00
Telex: 53.561

Honeywell AG
Computer Control Division
1200 Geneva/Switzerland
73, Route de Lyon
☎ 44 25 50
Telex: 22 670

## Other countries export

Honeywell Ltd.,
Computer Control Division,
53 Clarendon Road,
Watford, Herts.
☎ Watford 42391
Telex: 934227

THE QUEEN'S AWARD
TO INDUSTRY
1969

Designed and printed in U.K. by Copia Productum, Harrow